2020

On Patching Learning Discrepancies in Neural Network Training

Mohamed Elfeki
University of Central Florida

ON PATCHING LEARNING DISCREPANCIES IN NEURAL NETWORK TRAINING

by

MOHAMED ELFEKI
B.S. Faculty of Engineering, Alexandria University, 2016
M.Sc. Computer Science, University of Central Florida, 2018

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2020

Major Professor: Liqiang Wang

To my academic advisor, Dr. Liqiang Wang. In the many dark times, you were unconditionally supportive, and in the occasional breaks of light, you were noting but encouraging. I found you late in my PhD, but without you I could not have finished it.

To my dissertation committee: Dr. Ulas, Dr. Gita, and Dr. Mohamed. Through the endless interactions, collaborations and long brainstorming sessions, you were utterly helpful. May this dissertation prove that your time was not wasted on me. Without you, this PhD would have been of a substantially lower quality.

To my loving and tenacious mother. All the unexpected turns and disappointments in my journey have been a constant reminder of the hardships you've been through. Any good thing I've ever done was only because of you.

To my nieces and nephews. Without your stubbornness and ceaseless fights, I would have finished this dissertation two years ago. Please listen to your moms!!

To those who inspired, to those who supported, and to those who believed. This is mostly your work, even if it was inscribed by me.

Thank you so much for being there when I needed you the most.

And finally, to myself.

The only one who knew how hard this journey has been, yet never gave up.

ABSTRACT

Neural network's ability to model data patterns proved to be immensely useful in plethora of practical applications. However, using physical world's data can be problematic since it is often cluttered, crowded with scattered insignificant patterns, contain unusual compositions, and widely infiltrated with biases and imbalances. Consequently, training a neural network to find meaningful patterns in seas of chaotic data points becomes virtually as hard as finding a needle in a haystack. Specifically, attempting to simulate real-world multi-modal noisy distributions with high-precision leads the network to learning an ill-informed inference distribution. In this work, we discuss four techniques to mitigate common discrepancies between real-world representations and the training distribution learned by the network. Namely, we address the techniques of Diverse sampling, objective generalization, domain and task adaptation being introduced as priors in learning the primary objective. For each of these techniques, we contrast the basic training where no prior is applied to the learning with our proposed method and show the advantage of guiding the training distribution to the critical patterns in real-world data using our suggested approaches. We examine those discrepancy-mitigation techniques on a variety of vision tasks ranging from image generation and retrieval to video summarization and actionness ranking.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: MOTIVATION

One of the defining characteristics of intelligence is the ability of an agent to model their surrounding environment and make accurate future predictions. Specifically, an intelligent agent should be able to perform two primary tasks. First, the agent needs to construct an accurate representation of reality by eliminating noise and insignificant signal components from their sensory input. Second, the agent uses an inference framework to utilize the principal components of the supervisory or self-supervisory signal to extrapolate and predict the optimal set of future actions as constrained by an inference objective.

In the last few decades, neural networks have proven to be a remarkable approach to tackle the latter task(e.g., [146, 59, 169]). Their capacity to disclose latent patterns and provide an accurate classification or regression predictions have outsmarted significant number of other machine learning algorithms [79, 5], and often humans [72, 159]. Their power has been successfully applied to a variety of tasks starting from image understanding [157] to natural language processing [111]. However, their success on those tasks is primarily reliant on the quality of data samples that networks are learning. The provided training data points are sampled from distributions representing physical real-world. Thus, any defect or oversight that occur in the sampling procedure will affect the learning ability of the neural network and subsequently provide weak and inaccurate predictions.

In this work, our main purpose is to provide an insight to the former task: correctly modeling physical reality into an accurate information representation. The intricate nature of the physical nature entails that data used for training intelligent agents is predominantly noisy and often misleading. This composite nature of real-world information produces a disparity between what nature encompasses and what is being modeled by neural networks. Figure 1.1 shows few common dis-

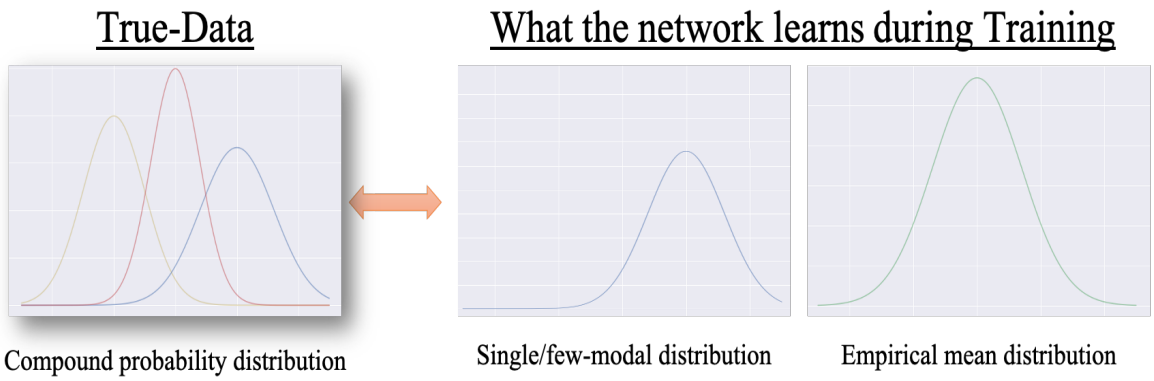

Figure 1.1: Some common discrepancies that occur in neural network training. While true-data is often a multi-modal mixture of distinct probability distributions, the network often learns only few modes or an empirical sample mean distribution that is potentially skewed from population mean distribution. Neither of those models is an accurate representation of real world's data.

crepancies between what the network learns and the true representation of reality. Specifically, we can model a generic task information signal as a Gaussian mixture model. A common discrepancy happens when the network only learns one or few modes of the mixture, which is discussed in Sec. 1.1. Another discrepancy is caused by the network learning a single empirical mean distribution instead of a multi-variate distribution. The empirical mean can be skewed based on how well the sampling procedure is performed. Thus, this case can be embodied in many situations including: insufficient samples (Sec. 1.2), complex nature of the target distribution (Sec. 1.3), or inadequate objective (Sec. 1.4).

1.1 Objective 1: Covering all true-data modes

Due to the composite nature of real-world data, their population distribution tends to be a combination of diverse distributions(i.e., data modes). Often, sampling true-data distribution tends to

overlook some of the data modes or give a higher weight to some on the expanse of others in terms of how many samples are representing each data mode. Thus, training data may encounter unbalanced data modes, or even fail to represent a given balanced distribution. Subsequently, a failed training can be caused by unbalanced data representation, or network's failure to represent all the modes in a balanced data samples, or a combination of both factors. In any of these cases, a successful data modeling needs to administer the network to model all critical patterns by diversifying data modes and learning the diversity structure of true-data representation. We examine this discrepancy by introducing a diversifying sampling procedures as a prior in training generative models.

1.2 Objective 2: Employing an Auxiliary Domain

Occasionally, the complexity of the target task may outweigh the representation capacity of the provided data. However, there is a multitude of options to train neural network with insufficient training samples. Some solutions may include using a shallower network with lower learning capacity to avoid overfitting, or using data augmentation to make the most out of the given data. In this chapter, we choose to focus on learning data sampled from a similar domain to improve the performance on the primary domain. This approach introduces cross-domain generalization capability in the learning, which assists in mitigating the risk of domain overfitting. We examine this concept by investigating the problem of cross-domain image retrieval and generation. We show that providing sufficient cross-domain training samples is an unattainable and often an infeasible task. And thus, we utilize data from an auxiliary domain to provide a prior in learning the hard problem with limited real-world training samples.

1.3 Objective 3: Employing an Auxiliary Task

Another complementary solution to target task complexity exceeding the modeling capacity is utilizing an auxiliary task to facilitate the learning procedure on the original task. Employing an assisting task to the learning of the primary target, not only provides a better task-generalization capability. But also it guides optimization to a more refined learning on the primary task, assuming the multi-task learning objective is fine-tuned properly. In this chapter, we select the video summarization as a non-trivial primary task and actionness ranking as a relatively simpler auxiliary task. We observe an imperative relationship between the two tasks. So we empirically prove the existence of such relationship between the two tasks and propose a framework to learn the primary task while generalizing and advancing across tasks.

1.4 Objective 4: Expanding objective's scope

The last discrepancy we explore is the limited-scope objective. Training a neural network under a poorly-defined target jeopardizes the integrity of the modeling capability. Thus, if the objective is defined only for a part of the true-data distribution but applied to samples drawn from out of the defined scope, the network fails to properly infer the correct predictions. In the final chapter of this work, we discuss the shortcomings of training neural networks with poorly defined objective on the problem of video summarization. Specifically, we propose a generalization for the particular problem of multi-view summarization under diversity constraint. We compare our generalized approach with the limited-scope objective to show the advantage of using a well-crafted, generalized-scope training objective as contrasted with the special-cases, ill-defined basic objective.

CHAPTER TWO: DIVERSE SAMPLING

GDPP: Learning Diverse Generations using Determinantal Point Processes

Mohamed Elfeki, Camille Couprie, Morgane Riviere, Mohamed Elhoseiny

Published in Thirty-sixth International Conference on Machine Learning (2019/6/13)

2.1 Abstract

Generative models have proven to be an outstanding tool for representing high-dimensional probability distributions and generating realistic looking images. An essential characteristic of generative models is their ability to produce multi-modal outputs. However, while training, they are often susceptible to mode collapse, that is models are limited in mapping input noise to only a few modes of the true data distribution. In this chapter, we draw inspiration from Determinantal Point Process (DPP) to propose an unsupervised penalty loss that alleviates mode collapse while producing higher quality samples. DPP is an elegant probabilistic measure used to model negative correlations within a subset and hence quantify its diversity. We use DPP kernel to model the diversity in real data as well as in synthetic data. Then, we devise an objective term that encourages generator to synthesize data with a similar diversity to real data. In contrast to previous state-of-the-art generative models that tend to use additional trainable parameters or complex training paradigms, our method does not change the original training scheme. Embedded in an adversarial training and variational autoencoder, our Generative DPP approach shows a consistent resistance to mode-collapse on a wide-variety of synthetic data and natural image datasets including MNIST, CIFAR10, and CelebA, while outperforming state-of-the-art methods for data-efficiency, generation quality, and convergence-time whereas being 5.8x faster than its closest competitor. [1]

[1] https://github.com/M-Elfeki/GDPP

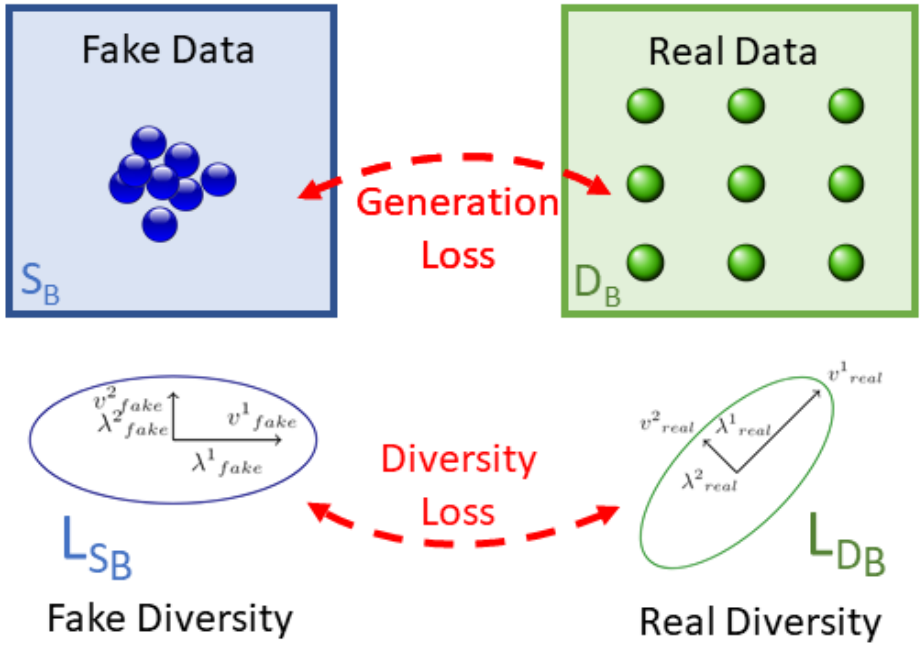

Figure 2.2: Inspired by DPP, we model a batch diversity using a kernel L. Our loss encourages generator G to synthesize a batch S_B of a diversity L_{S_B} similar to the real data diversity L_{D_B}, by matching their eigenvalues and eigenvectors. Generation loss aims at generating similar data points to the real, and diversity loss aims at matching the diversity manifold structures.

2.2 Introduction

Deep generative models have gained great research interest in recent years as a powerful framework to represent high dimensional data in an unsupervised fashion. Among many generative approaches, Generative Adversarial Networks (GANs) [43] and Variational AutoEncoders (VAEs) [71] took a place among the most prominent approaches for synthesizing realistic images. They consist of two networks: a generator (decoder) and a discriminator (encoder), where the generator attempts to map latent code to *fake* data points that simulate the distribution of *real* data. Nevertheless, in the process of learning multi-modal complex distributions, both models may converge to a trivial solution where the generator learns to produce few modes exclusively, which referred to by mode collapse.

To address this, we propose using Determinantal Point Processes (DPP) to model the diversity within data samples. DPP is a probabilistic model that has been mainly adopted for solving subset selection problems with diversity constraints [78], such as video and document summarization. In such cases, representative sampling requires quantifying the diversity of 2^N subsets, where N is the size of the ground set. However, this renders DPP sampling from true data to be computationally inefficient in the generation domain. The key idea of our work is to model the diversity within real and fake data throughout the training process using DPP kernels, which adds an insignificant computational overhead. Then, we encourage producing samples of similar diversity distribution to the true data by back-propagating our proposed DPP-inspired metric through the generator. In such a way, the generator explicitly learns to cover more modes of real distribution without a significant overhead.

Recent approaches tackled mode-collapse in one of two different ways: (1) modifying the learning of the system to reach a better convergence point (e.g. [110, 45]); or (2) explicitly enforcing the models to capture diverse modes or map back to the true-data distribution (e.g. [151, 13]). Here we focus on a relaxed version of the latter, where we use the same learning paradigm of the standard generators and add a penalty term to the objective function. The advantage of such an approach is to avoid adding any extra trainable parameters to the framework while maintaining the same back-propagation steps as the default learning paradigm. Thus, our model converges faster to a fair equilibrium point where generator imitates the diversity of true-data distribution and produces higher quality generations.

2.2.1 Contribution

We introduce a new penalty term, that we denote Generative Determinantal Point Processes (*GDPP*) loss. Our loss only assumes access to a generator G and a feature extraction function $\phi(\cdot)$. The

loss encourages the generator to diversify generated samples to match the diversity of real data as illustrated in Fig. 2.2. This criterion can be considered as a complement to the original generation loss which attempts to learn an indistinguishable distribution from the true-data distribution without explicitly enforcing diversity. We assess the performance of GDPP on three different synthetic data environments, while also verifying its advantage on three real-world images datasets. Our approach consistently outperforms several state-of-the-art approaches that of more complex learning paradigms in terms of alleviating mode-collapse and generation quality.

2.3 Related Work

Among many existing generation frameworks, GANs tend to synthesize the highest quality generations, however, they are harder to optimize due to unstable training dynamics. Here, we discuss a few generic approaches addressing mode collapse with an emphasis on GANs. We categorize them based on their approaches to alleviate mode collapse.

2.3.1 Mapping generated data back to noise

[23, 24] are of the earliest methods that proposed learning a reconstruction network besides learning the generative network. Adding this extra network to the framework aims at reversing the action of generator by mapping from data to noise. Likelihood-free variational inference (LFVI) [161], merges this concept with learning implicit densities using hierarchical Bayesian modeling. Ultimately, VEEGAN [151] used the same concept, but without basing reconstruction loss on the discriminator. This has the advantage of isolating the generation process from the discriminator's sensitivity to any of the modes. Along similar lines, [13] proposed several ways of regularizing the objective of adversarial learning including geometric metric regularizer, mode regularizer, and

manifold-diffusion training. Specifically, mode regularization has shown a potential into alleviating mode collapse and stabilizing the training.

2.3.2 Providing a surrogate objective function

InfoGAN [17] propose an information-theoretic extension of GANs that obtains disentangled representation of data by latent-code reconstitution through a penalty term in its objective. InfoGAN includes autoencoder over latent codes; however, it was shown to have stability problems similar to the standard GAN and requires stabilization empirical tricks. The Unrolled-GAN of [110] propose a novel objective to update the generator with respect to the unrolled optimization of the discriminator. This allows training to be adjusted between using the optimal discriminator in the generator's objective, which has been shown to improve the generator training process and to reduce mode collapse. Generalized LS-GAN of [26] define a pullback operator to map generated samples to the data manifold. With a similar philosophy, BourGAN [168] draws samples from a mixture of Gaussians instead of a single Gaussian. There is, however, no specific enforcement to diversify samples. Finally, improving Wasserstein GANs of [8], WGAN-GP [45] introduce a gradient penalization employed in state-of-the-art systems [68].

2.3.3 Using multiple generators and discriminators

One of the popular methods to reduce mode collapse is using multiple generator networks to provide better coverage of the true data distribution. [94] propose using two generators with shared parameters to learn the joint data distribution. The two generators are trained independently on two domains to ensure a diverse generation. However, sharing the parameters guide both the generators to a similar subspace. [25] propose a similar idea of multiple discriminators that are being an ensemble, which was shown to produce better quality samples. Recently, [39] proposed MAD-GAN

which is a multi-agent GAN architecture incorporating multiple generators and one discriminator. Along with distinguishing real from fake samples, the discriminator also learns to identify the generator that synthesized the fake sample. The learning of such a system implies forcing different generators to learn unique modes, which helps in better coverage of data modes. DualGAN of [115] improves the diversity within GANs at the additional requirement of training two discriminators. The Mixed GAN approach of [97] rather introduces a permutation invariant architecture for the discriminator, that doubles the number of parameters. In contrast to these approaches, our GDPP-GAN does not require any extra trainable parameters which results in a faster training as well as being less susceptible to overfitting.

Finally, we also refer to PacGAN [91] which modifies the discriminator input with concatenated samples to better sample the diversity within real data. Nevertheless, such an approach is subject to memory and computational constraints as a result of the significant increase in batch size. Additionally, spectral normalization strategies have been recently proposed in [113] and SAGAN [177] to further stabilize the training. We note that these strategies are orthogonal to our contribution and could be implemented in conjunction with ours to further improve the training stability of generative models.

2.4 Determinantal Point Process (DPP)

DPP is a probabilistic measure was introduced in quantum physics [102] to model the Gauss-Poisson and the 'fermion' processes, then was extensively studied in random matrix theory, e.g. [60]. It provides a tractable and efficient means to capture negative correlation with respect to a similarity measure, that in turn can be used to quantify the diversity within a subset. As pointed out by [42], DPP is agnostic about the order of the items within subsets. Hence, it can be used to model data that is randomly sampled from a certain distribution such as mini-batches sampled

from training data.

A point process $\mathcal{P}$ on a ground set $\mathcal{V}$ is a probability measure on the power set 2^N, where $N = |\mathcal{V}|$ is the size of the ground set. A point process $\mathcal{P}$ is called determinantal if, given a random subset Y drawn according to $\mathcal{P}$, we have for every $S \subseteq Y$,

$$\mathcal{P}(S \subseteq Y) \propto \det(L_S) \tag{2.1}$$

for some symmetric similarity kernel $L \in \mathbb{R}^{N \times N}$, where L_S is the similarity kernel of subset S. L must be real, positive semidefinite matrix $L \preceq I$ (all the eigenvalues of L are between 0 and 1); since it represents a probabilistic measure and all of its principal minors must be non-negative.

L is often referred to as the marginal kernel because it contains all the information needed to compute the probability of any subset S being selected in $\mathcal{V}$. L_S denotes the sub-matrix of L indexed by S, specifically, $L_S \equiv [L_{ij}]; i, j \in S$. Hence, the marginal probability of including one element e_i is $p(e_i \in Y) = L_{ii}$, and two elements e_i and e_j is $L_{ii}L_{jj} - L_{ij}^2 = p(e_i \in Y)p(e_j \in Y) - L_{ij}^2$. A large value of L_{ij} reduces the likelihood of both elements to appear together in a diverse subset.

[75] proposed decomposing the kernel L_S as a Gram matrix:

$$\mathcal{P}(S \subseteq Y) \propto \det(\phi(S)^\top \phi(S)) \prod_{e_i \in S} q^2(e_i), \tag{2.2}$$

where $q(e_i) \geq 0$ can be seen as a quality score of an item e_i in the ground set $\mathcal{V}$, while $\phi_i \in \mathcal{R}^D; D \leq N$ and $||\phi_i||_2 = 1$ is used as an ℓ_2 normalized feature vector of an item. In this manner, $\phi_i^\top \phi_j \in [-1, 1]$ is evaluated as a "normalized similarity" between items e_i and e_j of $\mathcal{V}$, and the kernel L_S is guaranteed to be real positive semidefinite matrix.

2.4.1 Geometric interpretation

$\det(\phi(S)^\top \phi(S)) = \prod_i \lambda_i$, where λ_i is the i^{th} eigen value of the kernel $\phi(S)^\top \phi(S)$, and $\lambda \geq 0$ since the kernel is a positive semidefinite matrix. Hence, we may visualize that DPP models diverse representations of data because the determinant of $\phi(S)^\top \phi(S)$ corresponds to the volume in N-D which is equivalent to the multiplication of data variances (i.e., the eigen values).

2.4.2 DPP in literature

DPP has proven to be a valuable tool when tackling diversity enforcement in problems such as document summarization (e.g., [78, 58]), pose estimation (e.g., [47]) and video summarization (e.g., [42, 105]). For instance, [179] proposed to learn the two parameters q, ϕ in eq. 5.18 to quantify the diversity of the kernel L_S based on spatio-temporal features of the video to perform summarization. Recently, [61] proposed to use DPP to automatically create capsule wardrobes, i.e. assemble a minimal set of items that provide maximal mix-and-match outfits given an inventory of candidate garments.

2.5 Approach

Our GDPP loss encourages the generator to sample fake data of diversity similar to real data diversity. The key challenge is to model the diversity within real data and fake data. We discussed in Sec. 2.4 how DPP can be used to quantify the diversity within a discrete data distribution. Unlike subset selection problems (e.g., document/video summarization), in the generation domain we are not merely interested in increasing diversity within generated samples. Only increasing the samples diversity will result in samples that are far apart in the generation domain, but not necessarily representative of real data diversity. Instead, we aim to generate samples that imitate

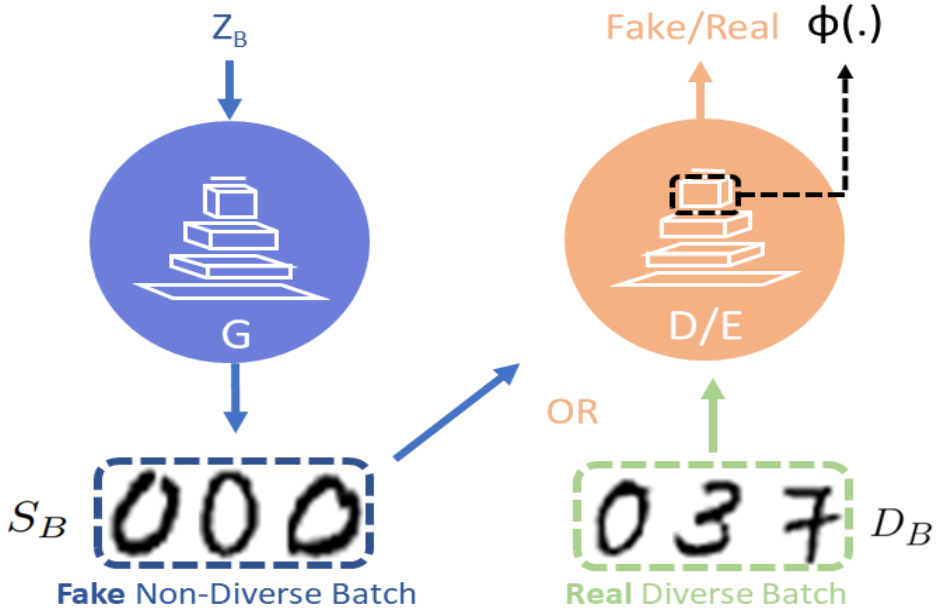

Figure 2.3: Given a generator G and feature extraction function $\phi(\cdot)$, the diversity kernel is constructed as $L = \phi^\top \cdot \phi$. By modeling the diversity of fake and real batches, our loss matches their kernels L_{S_B} and L_{D_B} to encourage synthesizing samples of similar diversity to true data. We use the last feature map of the discriminator in GAN or the encoder in VAE as the feature representation ϕ.

the diversity of real data. Thus, we construct a DPP kernel for both the real data and the generated samples at every iteration of the training process as shown in Fig. 2.3. Then, we encourage the generator to synthesize samples that have a similar diversity kernel to that of the training data. In order to simplify learning kernels, we match the eigenvalues and eigenvectors of the fake data DPP kernel with their corresponding of the real data DPP kernel. Eigenvalues and vectors capture the manifold structure of both real and fake data, and hence renders the optimization more feasible. Fig. 2.2 shows pairing the two kernels by matching their high dimensional eigen manifolds.

During training, a generative model G produces a batch of samples $S_B = \{e_1, e_2, \cdots e_B\}; S_B = G(z_B)$, where B is the batch size and $z_B \in R^{d_z \times B}$ is noise vector inputted to the generator G. At every iteration, we also have a batch of samples $D_B \sim p_d$, where p_d is a sampler from true

distribution. Our aim is to produce S_B that is probabilistically sampled following the DPP kernel of D_B, which satisfies:

$$\mathcal{P}(S_B \subseteq Y) \propto \det(L_{D_B}) \tag{2.3}$$

such that Y is a random variable representing a fake subset S_B drawn with a generative point process $\mathcal{P}$, and L_{D_B} is DPP kernel of a real subset indexed by D_B.

To construct L_{S_B}, L_{D_B}, we use the kernel decomposition in Eq. 5.18. However, since both true and fake samples are drawn randomly with no quality criteria, it is safe to assume $q(e_i) = 1; \forall i \in 1, 2, ..., B$. Thus, we construct the kernels as follows: $L_{S_B} = \phi(S_B)^\top \phi(S_B)$ and $L_{D_B} = \phi(D_B)^\top \phi(D_B)$, such that $\phi(S_B)$ and $\phi(D_B)$ are feature representations extracted by the feature extraction function $\phi(\cdot)$.

Our aim is to learn a fake diversity kernel L_{S_B} close to the real diversity kernel L_{D_B}. Nonetheless, matching two kernels is an unconstrained optimization problem as pointed out by [84]. So, instead, we match the kernels using their major characteristics: eigenvalues and eigenvectors. This results in scaling down the matching problem into regressing the magnitudes of eigenvalues and the orientations of eigenvectors. Hence, our devised GDPP loss is composed of two components: diversity magnitude loss $\mathcal{L}_m$, and diversity structure loss $\mathcal{L}_s$ as follows:

$$\begin{aligned} &\mathcal{L}^{DPP} = \mathcal{L}_m + \mathcal{L}_s = \\ &\sum_i \|\lambda^i_{real} - \lambda^i_{fake}\|_2 - \sum_i \hat{\lambda}^i_{real} \cos(v^i_{real}, v^i_{fake}) \end{aligned} \tag{2.4}$$

where λ^i_{fake} and λ^i_{real} are the i^{th} eigenvalues of L_{D_B} and L_{S_B} respectively.

Finally, we account for the outlier structures by using the min-max normalized version of the eigenvalues $\hat{\lambda}^i_{real}$ to scale the cosine similarity between the eigenvectors v^i_{fake} and v^i_{real}. This aims

to alleviate the effect of noisy structures that intrinsically occur within the real data distribution or within the learning process.

2.5.1 Integrating GDPP loss with GANs

As a primary benchmark, we integrate our GDPP loss with GANs . Since our aim is to avoid adding any extra trainable parameters, we utilize features extracted by the discriminator: we choose to use the hidden activations before the last layer as our feature extraction function $\phi(.)$. We apply ℓ_2 normalization on the obtained features that guarantees constructing a positive semi-definite matrix according to eq. 2.4. We finally integrate $\mathcal{L}^{DPP}$ into the GAN objective by only modifying the generator loss of the standard adversarial loss [43] as follows:

$$\mathcal{L}_g = \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] + \mathcal{L}^{DPP} \tag{2.5}$$

2.5.2 Integrating GDPP loss with VAEs

A key property of our loss is its generality to any generative model. We show that by also embedding it within VAEs. A VAE consists of an encoder network $q_{\theta_1}(z|x)$, where x is an input training batch and z is sampled from a normal distribution parametrized by encoder outputs σ and μ, representing respectively the standard deviation and the mean of the distribution. Additionally, VAE has a decoder network $p_{\theta_2}(x|z)$ which reconstructs $\hat{x}$. We use the final hidden activations in q as our feature extraction function $\phi(.)$. Given a z sampled from a normal distribution $z \sim \mathcal{N}(\mu, \sigma)$, $p_{\theta_2}(\hat{x}|z)$ is used to generate the fake batch S_B, while the real batch D_B is randomly sampled from

Table 2.1: Degree of mode collapse and sample quality on mixtures of Gaussians. GDPP-GAN consistently captures the highest number of modes and produces better samples.

	2D Ring		2D Grid		1200D Synthetic	
	Modes (Max 8)	% High Quality Samples	Modes (Max 25)	% High Quality Samples	Modes (Max 10)	% High Quality Samples
GAN [43]	1	99.3	3.3	0.5	1.6	2.0
ALI [24]	2.8	0.13	15.8	1.6	3	5.4
Unrolled GAN [110]	7.6	35.6	23.6	16.0	0	0.0
VEE-GAN [151]	**8.0**	52.9	24.6	40.0	5.5	28.3
WGAN-GP [45]	6.8	59.6	24.2	28.7	6.4	29.5
GDPP-GAN	**8.0**	**71.7**	**24.8**	**68.5**	**7.4**	**48.3**

training data. Finally, we compute the $\mathcal{L}^{DPP}$ as in Eq. 2.4, rendering the GDPP-VAE loss as:

$$\begin{aligned}\mathcal{L}_{VAE} = &- \mathbb{E}_{z\sim q(z|x)}[\log\{p(x|z)\}] \\ &+ KL[q(z|x)||p(z)] + \mathcal{L}^{DPP}.\end{aligned} \tag{2.6}$$

2.6 Experiments

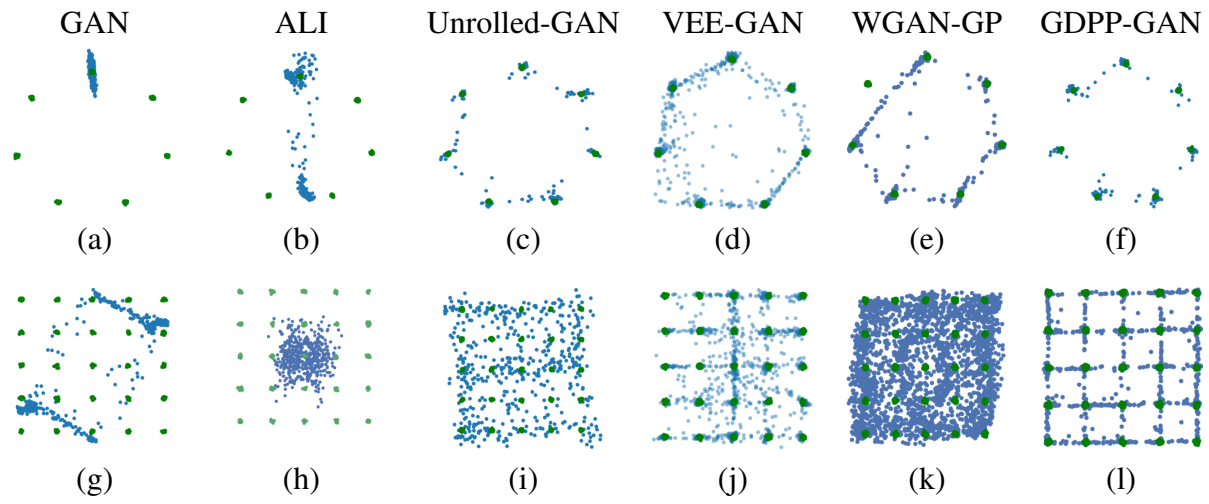

Figure 2.4: Scatter plots of the true data (green dots) and generated data (blue dots) from different GAN methods trained on mixtures of 2D Gaussians arranged in a ring (top) or a grid (bottom).

In our experiments, we target evaluating the generation based on two criteria: mode collapse and generated samples quality. Due to the intractability of log-likelihood estimation, this problem is non-trivial for real data. Therefore, we start by analyzing the performance on synthetic data where we can accurately evaluate these criteria. Then, we demonstrate the effectiveness of our method on real data using standard evaluation metrics. The same architecture is used for all methods and hyperparameters were tuned separately for each approach to achieve the best performance (See Appendix A.1.2 for details).

2.6.1 Synthetic Data Experiments

Mode collapse and the quality of generations can be explicitly evaluated on synthetic data since the true distribution is well-defined. In this section, we evaluate the performance of the methods on mixtures of Gaussian of known mode locations and distribution (See Appendix A.2 for details). We use the same architecture for all the models, which is the same one used by [110] and [151]. We note that the first four rows in Table 2.1 are obtained from [151], since we are using the same architecture and training paradigm. Fig. 2.4 illustrates the effect of each method on the 2D Ring and Grid data. As shown by the vanilla-GAN in the 2D Ring example (Fig. 2.4a), it can generate the highest quality samples however it only captures a single mode. On the other extreme, the WGAN-GP on the 2D grid (Fig. 2.4k) captures almost all modes in the true distribution, but this is only because it generates highly scattered samples that do not precisely depict the true distribution. GDPP-GAN (Fig. 2.4f,l) creates a precise representation of the true data distribution reflecting that the method learned an accurate structure manifold.

Table 2.2: GDPP loss Ablation study on GAN. $\mathcal{L}_s^u$ is the same as $\mathcal{L}_s$ without min-max eigen value normalization.

	2D Ring		2D Grid	
	Modes (Max 8)	% High Quality Samples	Modes (Max 25)	% High Quality Samples
Exact determinant ($\det[L_{S_B}]$)	**8**	**82.9**	12.6	21.7
Only diversity magnitude ($\mathcal{L}_m$)	**8**	67.0	20.4	15.9
Only diversity structure ($\mathcal{L}_s$)	**8**	65.2	18.2	35.2
GDPP with unnormalized structure term ($\mathcal{L}_m + \mathcal{L}_s^u$)	7.2	81.2	20.6	**68.8**
Final GDPP-loss ($\mathcal{L}_m + \mathcal{L}_s$)	**8**	71.7	**24.8**	68.5

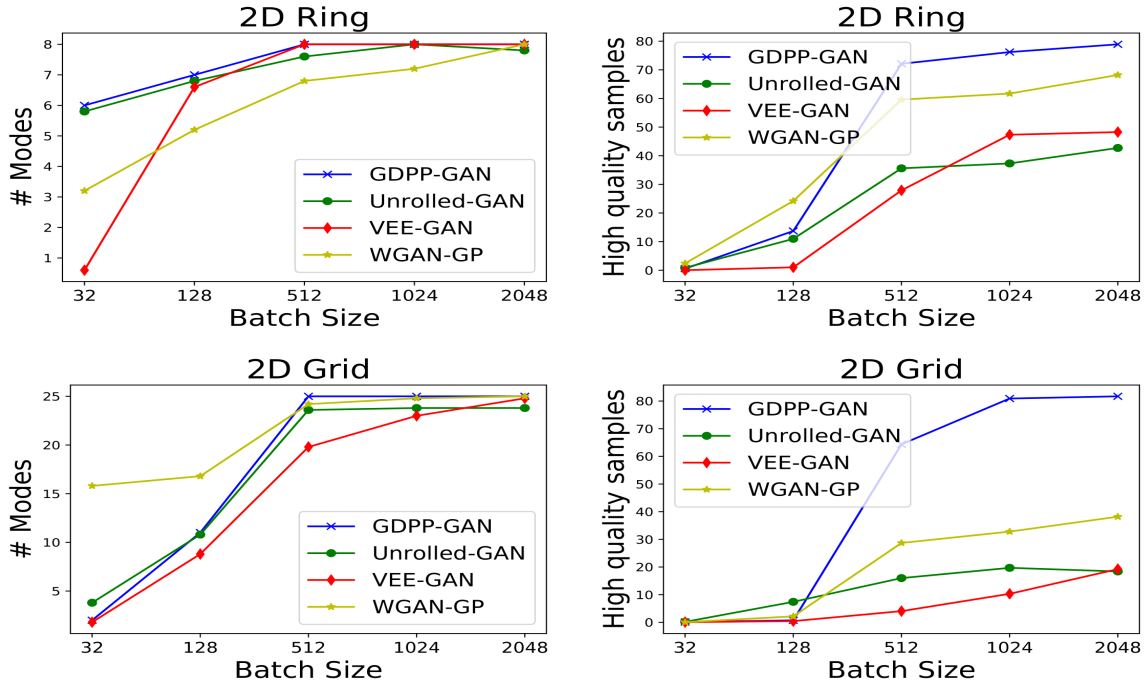

Figure 2.5: Data-Efficiency: examining the effect of training batch size B given the same number of training iterations. GDPP-GAN requires the least amount of training data to converge.

2.6.1.1 Performance Evaluation

At every iteration, we sample fake points from the generator and real points from the given distribution. Mode collapse is quantified by the number of real modes recovered in fake data, and the

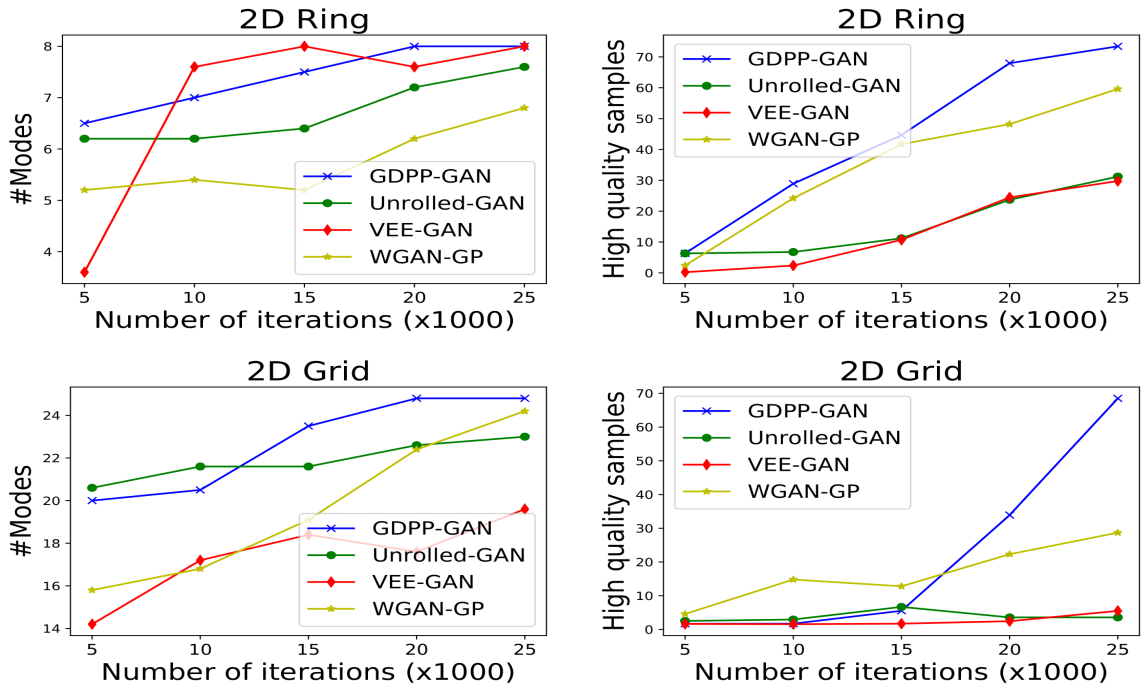

Figure 2.6: Time-Efficiency: monitoring convergence rate throughout the training given the same training data size. GDPP-GAN is the first to converge in both evaluation metrics.

generation quality is quantified by the % of High-Quality Samples. A generated sample is counted as high-quality if it was sampled within three standard deviations in case of 2D Ring or Grid, and ten standard deviations in case of the 1200D data. We train all models for 25K iterations, except for VEEGAN which needs 100K iterations to properly converge. At inference time, we generate 2500 samples from each of the trained models and measure both metrics. We report the numbers averaged over five runs with different random initialization in Table 2.1. GDPP-GAN clearly outperforms all other methods, for instance on the most challenging 1200D dataset that was designed to mimic a natural data distribution, bringing a 63% relative improvement in high-quality samples and 15% in mode detection over its best competitor WGAN-GP.

Finally, we show that our method is robust to random initialization. Since the weights of the generator are being initialized using a random number generator $\mathcal{N}(0, 1)$, the result of a generative

model may be affected by poor initializations. In Figure 7.30 we show qualitative examples on 2D Grid data, where we use high standard deviation for the random number generator $(i.e., \sigma > 100)$ as an example of poor initializations. Evidently, GDPP-GAN attains the true-data structure manifold even with poor initializations. On the other extreme, WGAN-GP tends to map the input noise to a disperse distribution covering all modes but with low-quality generations.

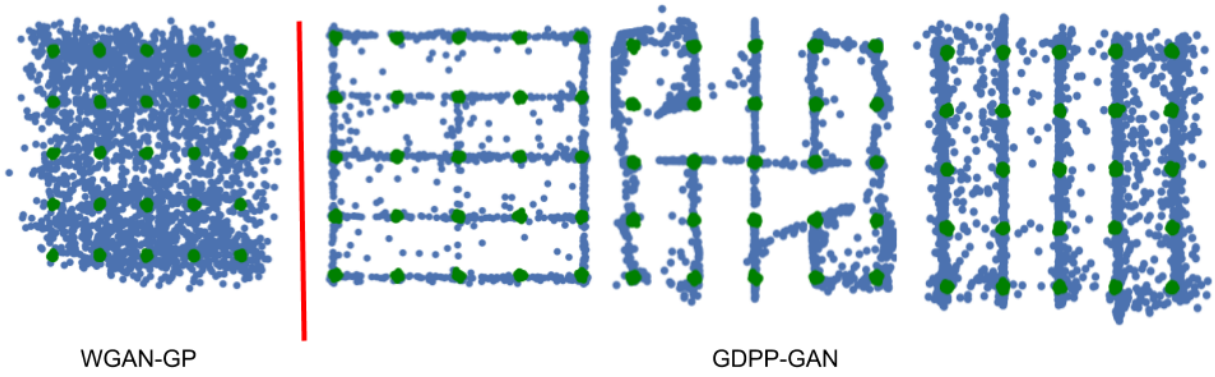

Figure 2.7: The effect of poor initialization on generations: GDPP-GAN models true manifold structure even with poor initializations, while WGAN-GP maps noise to disperse distribution covering the modes with low quality samples.

2.6.1.2 Ablation Study

We run a study on the 2D Ring and Grid data to show the individual effects of each component in our loss. As shown in Table 2.2, optimizing the determinant $\det L_S$ directly increases the diversity generating the highest quality samples. This works best on the 2D Ring since the true data distribution can be represented by a repulsion model. However, for more complex data as in 2D Grid, optimizing the determinant fails because it does not well-represent the real manifold structure but aims at repelling the fake samples from each other. Using GDPP with an unnormalized structure term $\mathcal{L}_s^u$ is prone to learning outlier caused by the inherent noise within the data. Nonetheless,

Table 2.3: Performance of various methods on real datasets. Stacked-MNIST is evaluated using the number of captured modes (Mode Collapse) and KL-divergence between the generated class distribution and true class distribution (Quality of generations). CIFAR-10 is evaluated by Inference-via-Optimization (Mode-Collapse) and Inception-Score (Quality of generations).

	Stacked-MNIST		CIFAR-10	
	#Modes (Max 1000)	KL div.	Inception score	IvO
DCGAN [132]	427	3.163	5.26 ± 0.13	0.0911
DeLiGAN [48]	767	1.249	5.68 ± 0.09	0.0896
Unrolled-GAN [110]	817	1.430	5.43 ± 0.21	0.0898
RegGAN [13]	955	0.925	5.91 ± 0.08	0.0903
WGAN [8]	961	0.140	5.44 ± 0.06	0.0891
WGAN-GP [45]	995	0.148	6.27 ± 0.13	0.0891
GDPP-GAN (Ours)	**1000**	**0.135**	$\mathbf{6.58 \pm 0.10}$	**0.0883**
VAE [71]	341	2.409	1.19 ± 0.02	0.543
GDPP-VAE (Ours)	**623**	**1.328**	$\mathbf{1.32 \pm 0.03}$	**0.203**

scaling the structure loss by the true-data eigenvalues $\hat{\lambda}$ seems to disentangle the noise from the prominent structure and better models the data diversity.

2.6.1.3 Data-Efficiency

We evaluate the amount of training data needed by each method to reach the same local optima as evaluated by our two metrics on both the 2D Ring and Grid data. Since the true-data is sampled from a mixture of Gaussians, we can generate an infinite size of training data. Therefore, we can quantify the amount of the training data by using the batch-size while fixing the number of back-propagation steps. In this experiment (Fig. 2.5), we run all the methods for the same number of iterations (25,000) and vary the batch size. However, WGAN-GP tends to capture higher quality samples with fewer data. In the case of 2D Grid data, GDPP-GAN performs on par with other methods for small amounts of data, yet it tends to significantly outperform other methods on the quality of generated samples once trained on enough data.

Table 2.4: Average Iteration running time on CIFAR-10. GDPP-GAN obtains the closest time to the default (non-improved) DCGAN.

	DCGAN	Unrolled-GAN	VEE-GAN	Reg-GAN	WGAN	WGAN-GP	GDPP-GAN
Avg. Iter. Time (s)	0.0674	0.2467	0.1978	0.1357	0.1747	0.4331	**0.0746**

2.6.1.4 Time-Efficiency

To analyze time efficiency, we explore two primary aspects: convergence rate, and physical running time. First, to find out which method converges faster, we fix the batch size at 512 and vary the number of training iterations for all models (Fig. 2.6). In the 2D Ring, only VEEGAN captures a higher number of modes before GDPP-GAN, however, they are of much lower quality than the ones generated by GDPP-GAN. In 2D Grid, however, GDPP-GAN performs on par with unrolled-GAN for the first 5,000 iterations while the others are falling behind. After then, our method significantly outperforms all the methods with respect to both the number of captured modes and the quality of generated samples. Second, we compare the physical running time of all methods given the same data and number of iterations. To obtain reliable results, we chose to run the methods on CIFAR-10 instead of the synthetic, since the latter has an insignificant running time. We compute the average running time of an iteration across 1000 iterations over five different runs of each method. Table 2.4 shows that GDPP-GAN has a negligible computational overhead beyond DCGAN, rendering it the fastest improved-GAN approach. We also elaborate on the run-time analysis and conduct additional experiments in Appendix A.3.3 to explore the computation overhead.

2.6.2 Image generation experiments

We run real-image generation experiments on three various datasets: Stacked-MNIST, CIFAR-10, and CelebA. For the first two, we use the experimental setting used in [45] and [110]. We also investigated the robustness of our method by using another more challenging setting proposed by [151] in Appendix A.3.2. For CelebA, we use the experimental setting of [68]. In our evaluation, we focus on comparing with the state-of-the-art methods that adopt a change in the original adversarial loss. Nevertheless, most baselines can be deemed orthogonal to our contribution and can enhance the generation if integrated with our approach. Finally, we show that our loss is generic to any generative model by incorporating it within Variational AutoEncoder (VAE) of [71] in Table 2.3. Appendix A.4 shows qualitative examples from several models and baselines.

2.6.2.1 Stacked-MNIST

A variant of MNIST [81] designed to increase the number of discrete modes in the data. The data is synthesized by stacking three randomly sampled MNIST digits along the color channel resulting in a $28 \times 28 \times 3$ image. In this case, Stacked MNIST has 1000 discrete modes corresponding to the number of possible triplets of digits. Following [45], we generate 50,000 images that are later used to train the networks. We train all the models for 15,000 iterations, except for DCGAN and unrolled-GAN that need 30,000 iterations to converge to a reasonable local-optima.

We follow [151] to evaluate the number of recovered modes and divergence between the true and fake distributions. We sample 26000 fake images for all the models. We identify the mode of each generated image by using the classifier mentioned in [13], which is trained on the standard MNIST dataset to classify each channel of the fake sample. The quality of samples is evaluated by computing the KL-divergence between generated label distribution and training labels distribution.

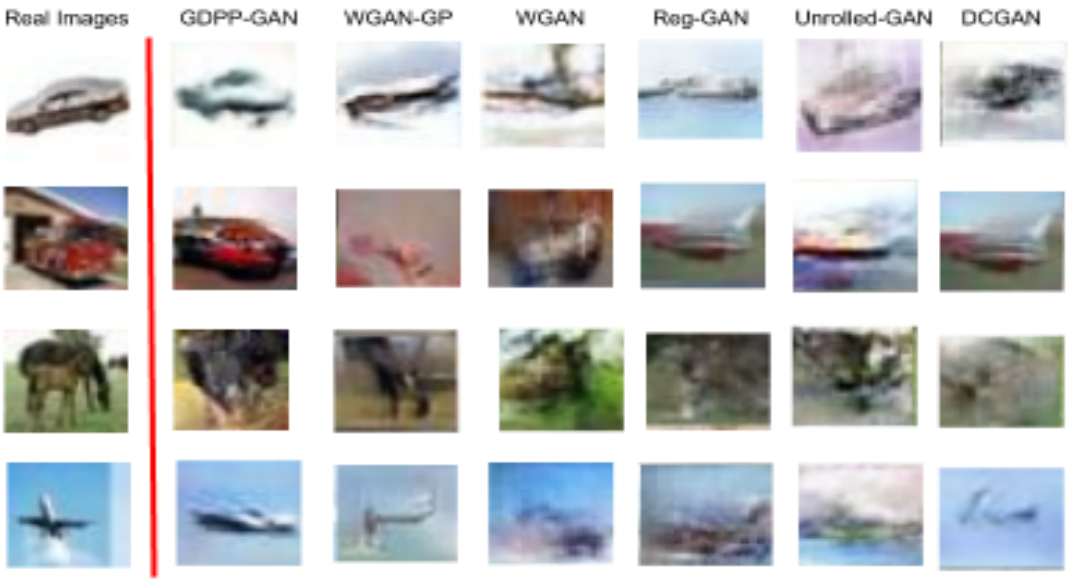

Figure 2.8: Real images and their nearest generations of CIFAR-10. Nearest generations are obtained by optimizing the input noise to minimize the reconstruction error of the generated image.

As shown in Table 2.3, GDPP-GAN captures all modes and generates a fake distribution that has the lowest KL-Divergence with the true-distribution. Moreover, when applied on the VAE, it doubles the number of modes captured (i.e., from 341 to 623) and cuts the KL-Divergence to half (from 2.4 to 1.3). Lastly, we follow [136] to assess the severity of mode collapse by computing the number of statistically different bins using MNIST in Appendix A.3.4.

2.6.2.2 CIFAR-10

We evaluate the methods on CIFAR-10 after training all the models for 100K iterations. Unlike Stacked-MNIST, the modes are intractable in this dataset. This is why we follow [110] and [151] in using two different metrics: Inception Score [141] for the generation quality and Inference-via-Optimization (IvO) for diversity. As shown in Table 2.3, GDPP-GAN consistently outperforms all other methods in both metrics. Furthermore, applying the GDPP on the VAE reduces the IvO by

63%. However, we note that both the inception-scores are considerably low which is also observed by [144] when applying the VAE on CIFAR-10.

Inference-via-optimization [110] is used to assess the severity of mode collapse in generations by comparing real images with the nearest generated image. In the case of mode collapse, there are some real images for which this distance is large. We measure this metric by sampling a real image x from the test set of real data. Then we optimize the ℓ_2 loss between x and generated image $G(z)$ by modifying the noise vector z. If a method attains low MSE, then it can be assumed that this method captures more modes than ones that attain a higher MSE. Fig. 2.8 presents some real images with their nearest optimized generations.

We also assess the stability of the training, by calculating inception score at different stages while training on CIFAR-10 (Fig. 2.9). Evidently, DCGAN has the least stable training with a high variation. However, by only adding GDPP penalty term to the generator loss, the model generates high-quality images the earliest on training with a stable increase.

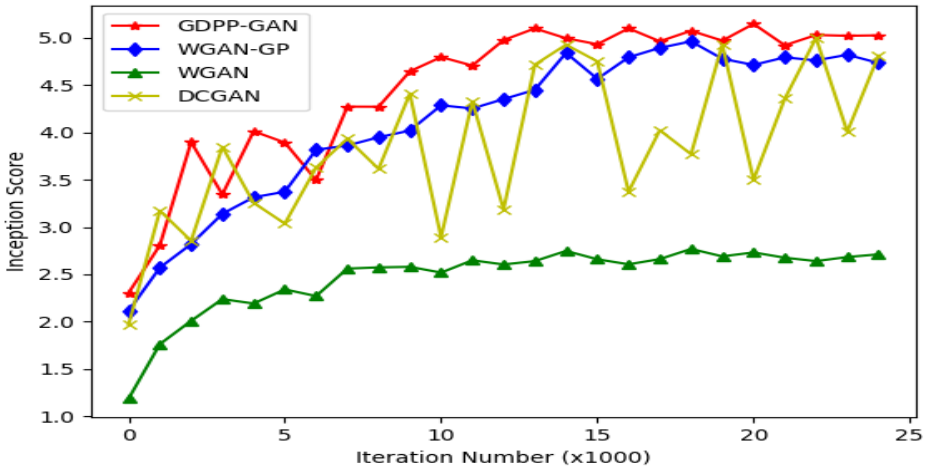

Figure 2.9: Adding GDPP loss to DCGAN stabilizes adversarial training and generates high quality samples earliest on CIFAR-10.

Table 2.5: Average and Minimum Sliced Wasserstein Distance over the last 10K iterations at scales 64^2, and scales 128^2 on CelebA. Training Data is the upper limit for this metric.

		Avg. SWD	Min. SWD
64×64	Training Data	0.0033	
	DCGAN	0.0906	0.0241
	WGAN-GP	0.0186	0.0115
	GDPP-GAN	**0.0163**	**0.0075**
128^2	Training Data	0.0023	
	WGAN-GP	0.0197	0.0095
	GDPP-GAN	**0.0181**	**0.0088**

2.6.2.3 CelebA

Finally, to evaluate the performance of our loss on large-scale adversarial training, we embed our GDPP loss in Progressive-Growing GANs [68]. We train the models for 40K iterations corresponding to 4 scales up to 64×64 results, and for 200K iterations at 5 scales (128×128). On large scale datasets such as CelebA dataset [95], it is harder to stabilize the training of DCGAN. In fact, DCGAN is only able to produce reasonable results in the first scale but not the second due to the high-resolution requirement. That is why, we embed our loss with WGAN-GP this time instead of DCGAN paradigm, which is as well orthogonal to our loss.

Unlike CIFAR-10 dataset, CelebA does not simulate ImageNet because it only contains faces, not natural scenes/objects. Therefore, using a model trained on ImageNet as a basis for evaluation (i.e., Inception Score), will cause inaccurate recognition. On the other hand, IvO was shown to be fooled by producing blurry images out of the optimization in high-resolution datasets as in CelebA [151]. Therefore, we follow [68] to evaluate the performance on CelebA using Sliced Wasserstein Distance (SWD) [128]. A small Wasserstein distance indicates that the distribution of the patches is similar, which entails that real and fake images appear similar in both appearance and variation at this spatial resolution. Accordingly, the SWD metric can evaluate the quality of

images as well as the severity of mode-collapse on large-scale datasets such as CelebA. Table 2.5 shows the average and minimum SWD metric across the last 10K training iterations. We chose this time frame because it shows a saturation in training loss for all the competing methods.

2.7 Conclusion

In this chapter, we introduced a novel criterion to train generative networks on capturing a similar diversity to one of the true data by utilizing Determinantal Point Process(DPP). We apply our criterion to Generative Adversarial training and the Variational AutoEncoder by learning a kernel via features extracted from the discriminator/encoder. Then, we train the generator on optimizing a loss between the fake and real, eigenvalues and eigenvectors of this kernel to encourage the generator on simulating the diversity of real data. Our GDPP framework accumulates many desirable properties: it does not require any extra trainable parameters, it operates in an unsupervised setting, yet it consistently outperforms state-of-the-art methods on a battery of synthetic data and real image datasets. Furthermore, GDPP-GANs exhibit a stabilized adversarial training and has been shown to be time and data efficient as compared to state-of-the-art approaches. Moreover, the GDPP criterion is architecture and model invariant, allowing it to be embedded with any variants of generative models such as adversarial feature learning and conditional GANs.

CHAPTER THREE: DOMAIN ADAPTATION

From Third Person to First Person: Dataset and Baselines for Synthesis and Retrieval

Mohamed Elfeki, Krishna Regmi, Shervin Ardeshir, and Ali Borji

Published in Computer Vision and Pattern Recognition EPIC (2019/6/20)

3.1 Abstract

In First-person (egocentric) and third person (exocentric) videos are drastically different in nature. The relationship between these two views have been studied in the recent years, however, it has yet to be fully explored. In this chapter, we introduce two datasets (synthetic and natural/real) containing simultaneously recorded egocentric and exocentric videos. We also explore relating the two domains (egocentric and exocentric) in two aspects. First, we synthesize images in the egocentric domain from the exocentric domain using a conditional generative adversarial network (cGAN). We show that with enough training data, our network is capable of hallucinating how the world would look like from an egocentric perspective, given an exocentric video. Second, we address the cross-view retrieval problem across the two views. Given an egocentric query frame (or its momentary optical flow), we retrieve its corresponding exocentric frame (or optical flow) from a gallery set. We show that using synthetic data could be beneficial in retrieving real data . We show that performing domain adaptation from the synthetic domain to the natural/real domain, is helpful in tasks such as retrieval. We believe that the presented datasets and the proposed baselines offer new opportunities for further research in this direction. The code and dataset are publicly available.[2]

[2]www.github.com/M-Elfeki/ThirdToFirst

3.2 Introduction

Recently egocentric cameras have gathered a plethora of data and have provided the opportunity to study first person vision extensively. At the same time, tremendous amount of research has been conducted on more traditional types of videos collected using static third-person cameras. We refer to these videos as exocentric. First-person and third-person domains, although drastically different, can be related together. In this chapter we take a step towards exploring this relationship. We are motivated by the fact that research in exocentric domain has a longer history relative to the first-person domain. Hence, there are more available datasets and benchmarks in this domain. Thus, effective transfer of information from third person to first person perspective could be very beneficial to research in the first-person domain. Understanding the relationship between these domains will facilitate exploiting existing models and solutions in exocentric domain and applying them to similar problems in egocentric domain.

Our contributions are three folds as explained below.

3.2.1 Dataset

We collect two datasets (synthetic and real), each containing simultaneously recorded egocentric and exocentric video pairs, where the egocentric is captured by body mounted cameras and the exocentric is captured by static cameras, capturing the egocentric camera holders performing diverse actions covering a broad spectrum of motions. We collect a large scale synthetic dataset generated using game engines, and provide frame level annotation on egocentric and exocentric camera poses, and the actions being performed by the actor. We also collect a smaller scale dataset of simultaneously recorded real egocentric and exocentric videos of actors performing different ac-

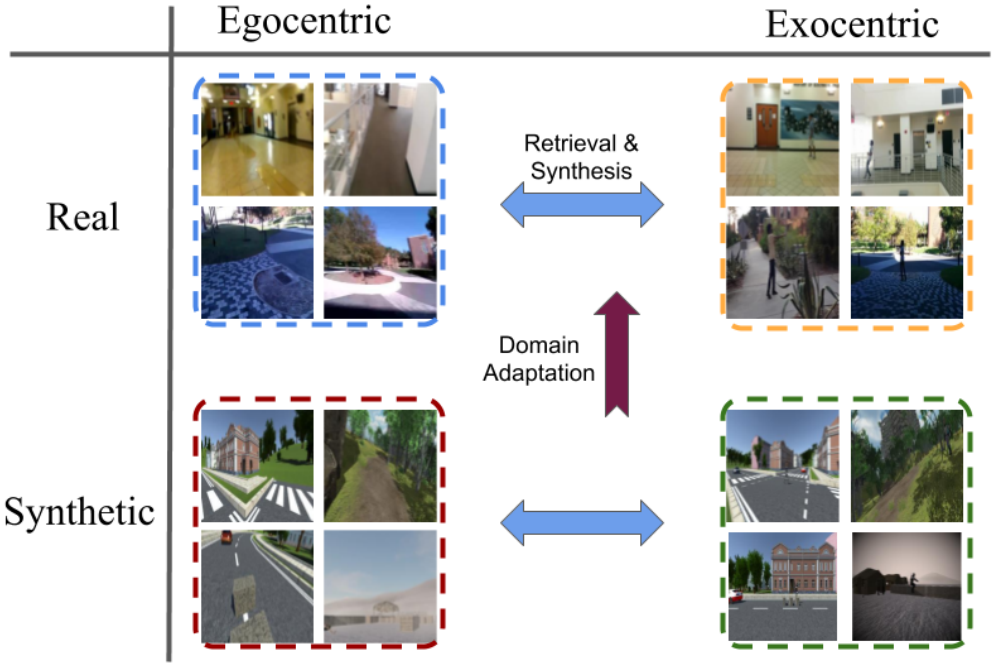

Figure 3.10: We study the relationship between first person and third person videos, in synthetic and natural domains. Domain adaptation from synthetic to real is helpful when we have limited real data, which is difficult to collect compared to synthetic data.

tions. We believe that the datasets and the annotations will be useful for exploring the relationship between first and third person videos in many aspects such as video retrieval and synthesis (as we explore here), action recognition, pose estimation, and 3D reconstruction. We believe that simultaneously recorded egocentric and exocentric videos could be beneficial in effectively exploring the relationship between these two domains, and could be benificial to the community.

3.2.2 Image Synthesis

Given an exocentric side-view image, we aim to generate an egocentric image hallucinating how the world would look like from a first person perspective. Synthesis is a very challenging computer

vision problem, especially when the generation is conditioned on images with drastically different views. In our work, the images in two domains often do not have a significant overlap in terms of their fields of view. Thus, transforming the appearances across the two views is non-trivial. As one of the contributions of this work, we attempt to address this problem across third person and first person images using conditional generative adversarial networks.

3.2.3 Retrieval

Given an exocentric frame in a video or its momentary optical flow (with respect to the previous frame), we explore retrieving its corresponding egocentric frame (or optical flow). To do so, we train a two stream convolutional neural network seeking a view invariant representation across the two views given a momentary optical flow map (a 2 channel input). We also train another network for RGB values (a 3 channel input). We perform domain adaptation across synthetic and real domain and show that using synthetic data improves the retrieval performance on real data.

3.3 Related Work

3.3.1 Egocentric Vision

First person vision, a.k.a egocentric vision, has become increasingly popular in the computer vision community. A lot of research has been conducted in the past few years [67, 9], including object detection [32], activity recognition [34, 33] and video summarization [96]. Motion in egocentric vision, in particular, has been studied as one of the fundamental features of first person video analysis. Costante et al. [19] explore the use of convolutional neural networks (CNNs) to learn the best visual features and predict the camera motion in egocentric videos. Su and Grauman

[155] propose a learning-based approach to detect user engagement by using long-term egomotion cues. Jayaraman et al. [63] learn the feature mapping from pixels in a video frame to a space that is equivariant to various motion classes. Ma et al. [99] have proposed a twin stream network architecture to analyze the appearance information and the motion information from egocentric videos and have used these features to recognize egocentric activities. Action and activity recognition in egocentric videos have been hot topics in the community. Ogaki et al. [119] jointly used eye motion and ego motion to compute a sequence of global optical flow from egocentric videos. Poleg et al. [129] proposed a compact 3D Convolutional Neural Network (3DCNN) architecture for long-term activity recognition in egocentric videos and extended it to egocentric video segmentation. Singh et al. [145] used CNNs for end-to-end learning and classification of actions by using hand pose, head motion and saliency map. Li et al. [89] used gaze information, in addition to these features, to perform action recognition. In their work, Matsuo et al. [108] have proposed an attention based approach for activity recognition by detecting visually salient objects.

3.3.2 Relating first and third person videos

The relationship between egocentric and top-view information has been explored in tasks such as human identification [1, 3, 30], semantic segmentation[5] and temporal correspondence[4]. In this work, we relate two different views of a motion, which can be considered as a knowledge transfer or domain adaptation task. Knowledge transfer has been used for the multi-view action recognition (e.g., [66, 92, 87]) in which multiple exocentric views of an action are related to each other. Having multiple exocentric views allows geometrical and visual reasoning, since: a) the nature of the data is the same in different views, and b) the actor is visible in all cameras. In contrast, our paper aims to automatically learn mappings between two drastically different views, egocentric and exocentric. To the best of our knowledge, this is the first attempt in relating these two domains for

transferring motion information. Cross-view relations have also been studied between egocentric (first person) and exocentric (surveillance or third-person) domains for action classification. [150] utilize the information from one egocentric camera and multiple exocentric cameras to solve the action recognition task, and [2] learns a mapping between first person and third person actions.

3.3.3 Generative Adversarial Networks

Goodfellow et al. [43] proposed the initial version of Generative Adversarial Networks for generating realistic images. Prior to that, Restricted Boltzmann Machines [53, 147] and deep Boltzmann Machines [140] have been used for that purpose. GANs have been used in conditional settings to synthesize images controlled by different parameters, such as labels of digits [112], images [62, 134, 135], textual descriptions [133, 176]. GANs are exploited for inpainting tasks by [126, 174]. We are the first to synthesize cross-view images involving egocentric and exocentric domains. In this work, we condition the generative adversarial networks on exocentric view image and attempt to hallucinate how the world looks from egocentric perspective.

3.4 Dataset

We collect a real dataset and a synthetic dataset containing simultaneously recorded egocentric and exocentric videos. In what follows, we briefly describe the two datasets and their statistics.

Table 3.6: Details of Real Dataset in terms of the number of training, validation and testing video and frame pairs.

	Training Pairs		Validation Pairs		Testing Pairs		Total Number of Pairs	
	#Vid	#Frames	#Vid	#Frames	#Vid	#Frames	#Vid	#Frames
Ego-Side	124	26,764	61	13,412	70	13,788	255	53,964
Ego-Top	135	28,408	68	12,904	73	14,064	276	55,376

3.4.1 Real Dataset

We present a dataset containing simultaneously recorded egocentric and exocentric videos covering a wide range of first and third person movements and actions. As this dataset is designed for studying the relationship between these two views, we isolate the egocentric camera holder in the third person video and thus, collect videos in which there is only a single person collecting an egocentric video and being recorded by an exocentric video. We collect a dataset containing 531 video pairs. Each video pair contains one egocentric and one exocentric (side or top-view) video. The pair of videos are temporally aligned, which will provide corresponding ego-exo image pairs. Some example frames are shown in Fig. 3.11. Each pair is collected by asking an actor to perform a range of actions (walking, jogging, running, hand waving, hand clapping, boxing, and push ups) covering a broad range of various motions and poses in front of an exocentric camera (top or side view), while wearing an egocentric body-worn camera capturing the actor's motion from the first person perspective. Details about the number of videos and statistics for training and testing are included in Table 3.6.

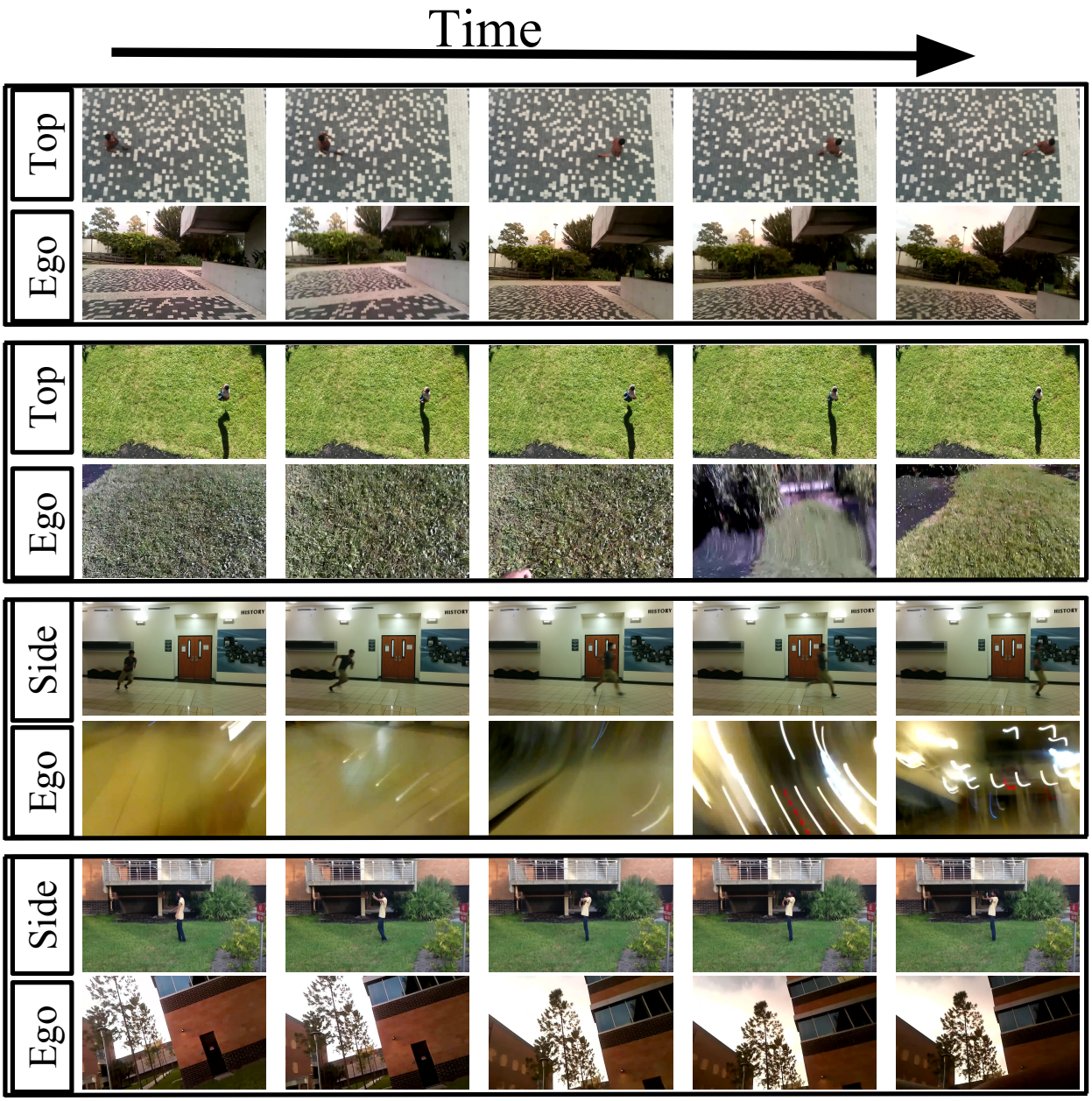

Figure 3.11: Examples from the real dataset: simultaneously recorded Ego-Top and Ego-Side pairs are shown.

3.4.1.1 Metadata and Annotations.

We provide frame level action labels for the videos in each view. Actions consist of walking, jogging, running, waving, boxing, clapping, jumping, and doing push-ups.

Table 3.7: Details of Synthetic Dataset in terms of the number of training, validation and testing video and frame pairs.

	Training Pairs		Validation Pairs		Testing Pairs		Total Number of Pairs	
	#Vid	#Frames	#Vid	#Frames	#Vid	#Frames	#Vid	#Frames
Ego-Side	208	119,115	109	6,702	95	6,778	412	132,595
Ego-Top	208	119,115	109	6,702	95	6,778	412	132,595

3.4.2 Synthetic Data

Since simultaneously recorded egocentric and exocentric videos are not abundant, collecting such data from the web and in large scale is not feasible. In order to attain a large number of samples, we collect a synthetic dataset using graphics engines. Several environments and actors were used in unity 3D platform, programmed to perform actions such as walking, running, jumping, crouching, etc. A virtual egocentric camera was mounted on the actor's body, while static virtual top/side view cameras were also positioned in the scene. We collected a large number of examples (more than 130,000 frames per camera) of such data. A few examples are shown in Fig. 3.12. In order to add variation to the data and make it resemble real data, we added slight random rotations to the virtual cameras. In our synthetic dataset, we have a total of 4 environments with 5, 7, 10 and 10 scenes. Scene refer to a location where the actions are recorded. For each environment, we use two scenes for testing and the rest for validation and training.

3.4.2.1 Metadata and Annotations

We provide frame level action labels, along with egocentric and exocentric camera poses. The action classes consist of walking, running, crouching, strafing, and jumping.

3.4.2.2 Dataset Value

We believe that the relationship across views (egocentric and exocentric) and modalities (synthetic and real data) could be explored in many aspects. Given that the dataset contains simultaneously recorded videos, and it contains frame level annotations in terms of action labels and camera poses, we believe that it could be used for many tasks such as video retrieval and video synthesis, for which we provide some baselines. Also this relationship could be explored in other tasks such as action recognition, camera pose estimation, human pose estimation, 3D reconstruction, etc.

3.5 Framework

3.5.1 Image Synthesis

Generative Adversarial Networks [43] are useful in synthesizing natural looking images which are not possible by minimizing the pixel-wise loss only during the training. GANs employ a generator network (G) that synthesizes the images very close to the training data distribution from noise distribution and a discriminator network (D) that is trained to discriminate between the samples generated by G and the original samples from the true data distribution. The discriminator acts as a learnable loss function to the generator to improve realism in synthesized images.

Conditional GANs use an auxiliary variable (e.g., labels [112], text embeddings [133, 176] or

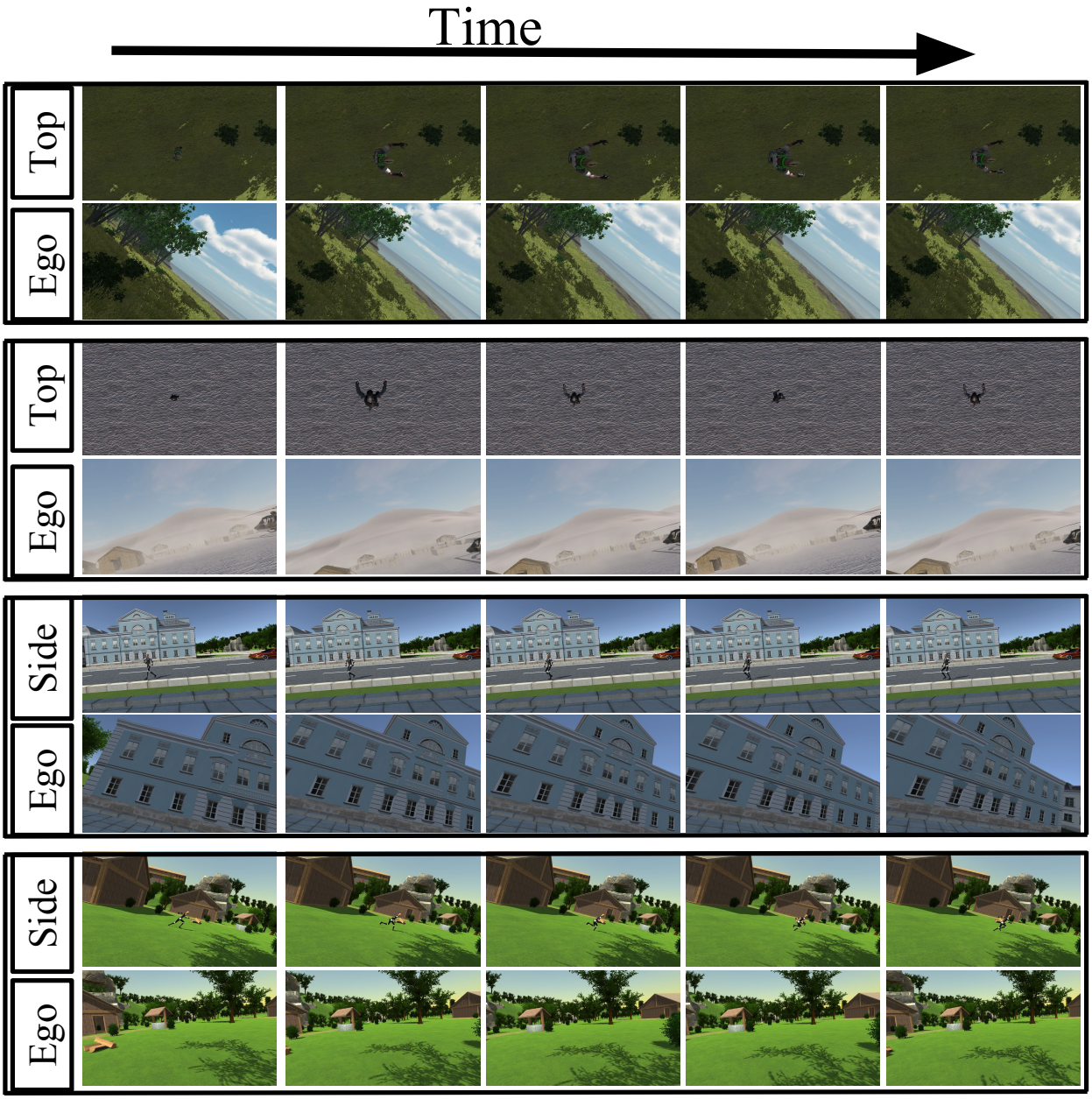

Figure 3.12: Examples from the synthetic dataset: simultaneously recorded Ego-Top and Ego-Side pairs are shown.

images [62, 134, 135, 188, 70]) as input to synthesize samples. Both G and D are shown the conditioning variable. G generates the target image using the auxiliary input. The conditioning

variable is paired with real/synthesized image and shown to D and D makes its prediction of whether the image pair it sees is real or fake.

Earlier works in GAN [62, 134, 126] used $L1$ or $L2$ distances between real and generated image pairs as additional term in loss function to encourage the generator to synthesize samples similar to the ground truth. Here, we use $L1$ distance as it increases image sharpness in the generation tasks.

In this work, we use an exocentric image (I_{exo}) as a conditional input to synthesize the ego image (I_{ego}). We minimize the adversarial loss and L1 loss during training. The conditional GAN loss and $L1$ loss are represented by Eq. (3.7) and Eq. (3.8), respectively.

$$\begin{aligned}\min_{\mathrm{G}}\max_{\mathrm{D}}\ L_{cGAN}(G,D) = E_{I_{ego},I_{exo}\sim p_{data}(I_{ego},I_{exo})}[logD(I_{ego},I_{exo})] \\ +E_{I_{exo},I'_{ego}\sim p_{data}(I_{exo},I'_{ego})}[log(1-D(I'_{ego},I_{exo}))],\end{aligned} \tag{3.7}$$

$$\min_{\mathrm{G}}\ L_{L1}(G) = E_{I_{ego},I'_{ego}\sim p_{data}(I_{ego},I'_{ego})}[||\ I_{ego} - I'_{ego}\ ||_1], \tag{3.8}$$

where, $I'_{ego} = G(I_{exo})$. The objective function for our network is the sum of conditional GAN loss in Eq. (3.7) and $L1$ loss in Eq. (3.8), as represented in Eq. (3.9):

$$L_{network} = L_{cGAN}(G,D) + \lambda L_{L1}(G), \tag{3.9}$$

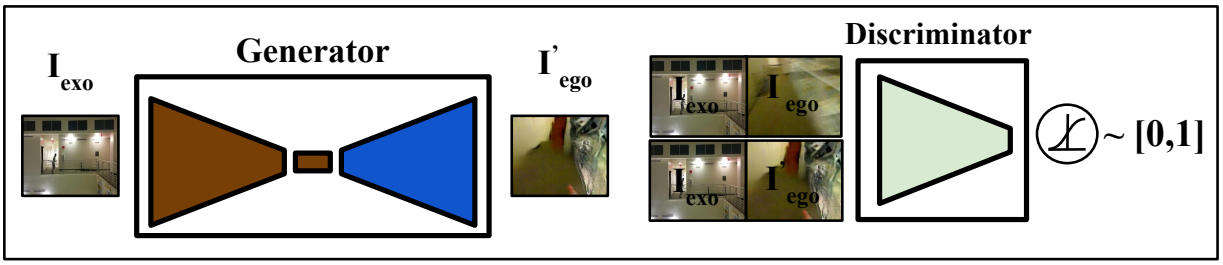

Figure 3.13: Image synthesis framework. An egocentric image is generated conditioned on an exocentric image. The exocentric image along with the real and synthesized egocentric images are passed to the discriminator as positive and negative pairs respectively.

where, λ is the balancing factor between the losses.

The architecture of our image generation network is shown in Fig. 3.13. I_{exo} is an exocentric image fed as a conditioning input to the network. The output of the generator is I'_{ego} which is the generated image in egocentric domain. The discriminator is provided with the (I_{exo}, I'_{ego}) pair as a negative example. The goal is to generate a I'_{ego} realistic enough to be able to fool the discriminator. The real image pair, (I_{exo}, I_{ego}) is also fed to the discriminator as a positive example.

We utilize the baseline model of [134] that was trained to generate street-view images from aerial images. We fine-tune the cross-view model for 15 epochs on our real and synthetic datasets. The images are first resized to 256 $\times$ 256 for generative tasks. We ran experiments with different hyperparameters but the ones from [134] worked best.

3.5.2 Retrieval

Given an egocentric video frame, we aim to retrieve its corresponding video frame across all the frames of all the exocentric videos. We perform retrieval based on the RGB values of the frames and also based on the optical flow. We perform retrieval using a two stream network with con-

trastive loss. We train a separate two stream network for RGB and one for Optical Flow. The architecture used for RGB based retrieval is shown in Fig. 3.14. We use the same architecture for retrieval based on momentary optical flow (optical flow at that specific time frame), with an exception to the number of input channels (3 for RGB and 2 for optical flow). We extract view specific features from each stream and encourage a view invariant embedding by setting the difference between corresponding pairs to zero.

3.5.2.1 Optical Flow:

We train a two stream network on the momentary optical flows extracted from each video. In others words, given a pair of simultaneously recorded exocentric and egocentric videos, we feed the optical flow at time t of the egocentric and exocentric video to the network as a positive pair. For any other pair of optical flow (frame t_1 in the egocentric and frame t_2 in exocentric where $t_1 \neq t_2$) the output of the network is set to 1 (negative pair). Since the optical flow maps are often very noisy, we perform a Gaussian smoothing over time in order to get more consistent flow maps, as a preprocessing step.

We train a network on the synthetic dataset (synthetic egocentric-exocentric pairs), and test it on the test set of the synthetic dataset. We perform the same experiment on the real dataset. We train and test another network on real dataset egocentric and exocentric pairs. We observe that the retrieval performance on the real data is not as favorable as the synthetic dataset, as the synthetic dataset is often less noisy, is in a more controlled environment, and has more training data. Given that the synthetic and real data are different in modality, we train a third retrieval network. We initialize the network with the weights trained on the synthetic dataset, and then fine-tune its convolutional layers on the synthetic data on the real data in order to benefit from the network pre-trained on the synthetic dataset. We observe that the retrieval performance of the fine-tuned network improves

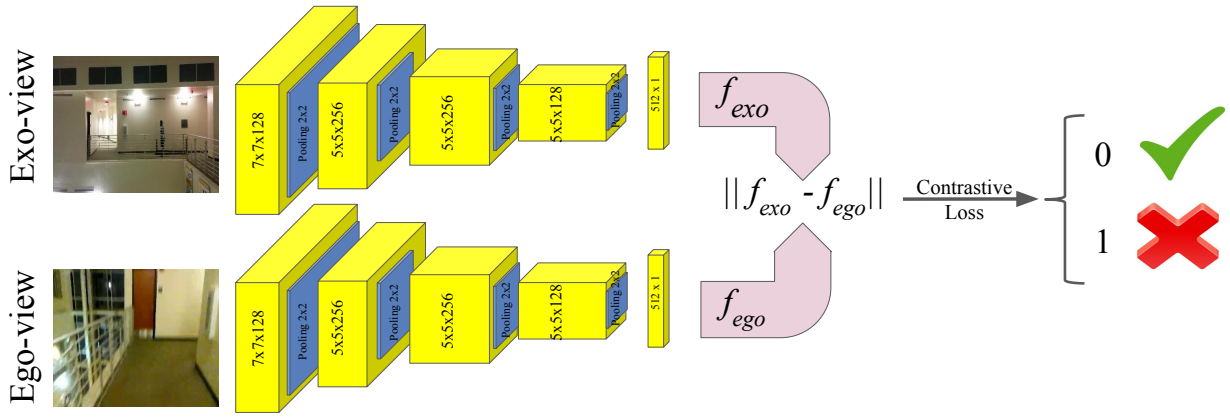

Figure 3.14: Retrieval Network Architecture.

significantly on the real data.

3.5.2.2 RGB

We perform the same experiments on the raw RGB values of the two views. We use the same structure as before and follow the same fine-tuning paradigm to ensure a better learning using the synthetic-trained weights on the real data. Our experiments show a substantial retrieval quality for both of the real as well as synthetic data. As before, the network that is pre-trained on the synthetic dataset and fine-tuned on the real dataset yields the best retrieval performance on the real dataset.

3.6 Experiments

3.6.1 Synthesis

A set of randomly selected qualitative results over real and synthetic datasets have been shown in Fig. 3.15. The generated frames show that the network is successful at transforming the semantic

information across the views. The generated images show blurriness for real dataset which is primarily because egocentric domain experiences motion in the frame rather than on the actor. The last two columns show some failure cases. The first failure case for real dataset shows the network is not able to learn the direction the person is facing so it is not able to generate the railings on right side of the person. The failure case for synthetic images show that the network is not able to hallucinate the textures in the scene. We use the following quantitative measures to evaluate the performance of the generated first person images:

Inception Score [160]: measures the diversity of the generated samples within a class, and their representative of the class. The inception score is computed as the following:

$$I = e^{E_x D_{KL}(p(y|x)||p(y))} \tag{3.10}$$

where x is a generated sample and y is its predicted label. We use the AlexNet model [73] trained on Places dataset [185] with 365 categories to compute the inception score for images. Following the [134], we also compute inception scores on Top-1 and Top-5 classes, where Top-k means that top k predictions for each image are unchanged while the remaining predictions are smoothed by an epsilon equal to $\frac{1-\Sigma top_k}{n-k}$.

3.6.1.1 Structural-Similarity (SSIM)

measures the similarity between the images based on their luminance, contrast and structural aspects. SSIM values range between -1 and +1. A higher value means greater similarity between the compared images. It is computed as

$$SSIM(I_{ego}, I'_{ego}) = \frac{(2\mu_{I_{ego}}\mu_{I'_{ego}} + c_1)(2\sigma_{I_{ego}I'_{ego}} + c_2)}{(\mu^2_{I_{ego}} + \mu^2_{I'_{ego}} + c_1)(\sigma^2_{I_{ego}} + \sigma^2_{I'_{ego}} + c_2)} \tag{3.11}$$

3.6.1.2 Peak Signal-to-Noise Ratio (PSNR)

measures the peak signal-to-noise ratio between two samples and evaluates the quality of the synthesized sample compared to the original sample. Higher values in PSNR imply better quality. It is computed as

$$PSNR(I_{ego}, I'_{ego}) = 10log_{10}(\frac{max^2 I'_{ego}}{\frac{1}{n}\Sigma_{i=0}^{n}(I_{ego}[i] - I'_{ego}[i])^2}) \tag{3.12}$$

where $maxI'_g = 255$ (maximum pixel intensity value).

3.6.1.3 Sharpness difference

similar to [107, 134], we compute the following:

$$SharpDiff(I_{ego}, I'_{ego}) = 10log_{10}(\frac{max^2 I'_{ego}}{\frac{1}{N}\Sigma_i\Sigma_j|(\nabla_i Y + \nabla_j Y) - (\nabla_i Y' + \nabla_j Y')|}) \tag{3.13}$$

where the denominator corresponds to the difference between the gradients of the generated and ground truth image. Intuitively, we would like the difference between the gradients to be small.

The inception scores are shown in Table 3.8. The higher inception scores for the real dataset

Table 3.8: Inception Scores for data and model distributions on Real and Synthetic Datasets.

Images	**Inception Score**		
	all classes	Top-1 class	Top-5 classes
Real Synthesized	3.8280	2.0315	3.4186
Real Ground-Truth	6.3787	2.6652	5.2608
Synthetic Synthesized	3.4320	2.1045	3.5042
Synthetic Ground-Truth	4.5353	2.3815	4.3695

Table 3.9: SSIM, PSNR and Sharpness Difference between real data and generated samples for Real and Synthetic Datasets.

Dataset	**SSIM**	**PSNR**	**Sharp Diff**
Real	0.4822	18.1694	19.8142
Synthetic	0.5153	20.8976	20.5758

is expected as the network was pretrained on natural images (Places dataset). SSIM, PSNR and Sharpness Difference scores are reported in Table 3.9. All of the scores are higher for the Synthetic dataset compared to the real dataset. This is mainly due to the fact that the synthetic dataset has a controlled environment with less motion blur compared to egocentric frames in real dataset.

3.6.2 Retrieval

We evaluate the retrieval performance using the cumulative matching curve (CMC). The area under curve (AUC) of the curves are used as a quantitative measure. We evaluate retrieval using optical flow, and report the results in Fig. 3.16 (left). We also illustrate the retrieval results based on RGB in Fig. 3.16 (right).

As explained before, we first train and test a two stream network on the synthetic dataset. The performance of this network is illustrated using the blue curve in Fig. 3.16, and is referred to as

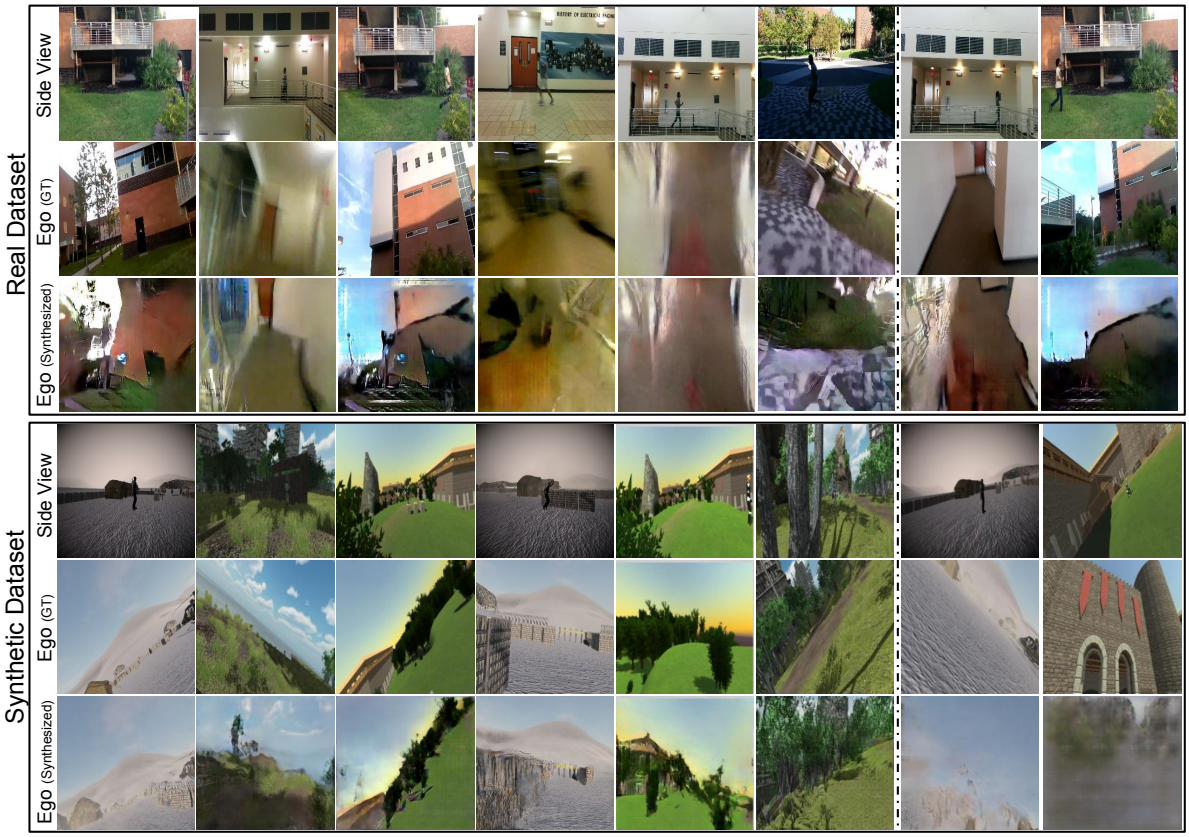

Figure 3.15: Qualitative Results for synthesis on Real (upper block) and Synthetic Datasets (lower block). In each block, first row shows images in exocentric (side) view, second row shows their corresponding ground truth egocentric images and the third row shows egocentric images generated by our method.

train S test S, where S stands for synthetic data. The green curve shows the performance of the two stream retrieval network trained and tested on the real data (train R test R, where R stands for real dataset). The red and blue curves are not directly comparable as they are tested on different datasets (synthetic and real). In general, the retrieval performance is not high over the real dataset due to its less amount of data and high noise. The orange curve (train S test R), shows the retrieval performance of the network trained on the synthetic data directly on the real data, which generally does not perform better than the network trained on the real data (green). However, once we fine-tune the network trained on the synthetic data on the real data, we attain better performance (red

curve, train S-R, test R). Except the blue curve which is tested on the synthetic data, all other curves are comparable as they have been tested on the real dataset. The best performance is achieved when the network is trained on synthetic data, and then its convolutional layers are tuned on the real data. The performance of chance (randomly ranking) is shown by the purple curve (chance).

3.6.2.1 Retrieval based on Optical Flow:

The cumulative matching curves for retrieval based on optical flow is shown in Fig. 3.16 (right). It can be observed that the network trained on synthetic and tested on real (orange) perform as chance level. The effect of adapting the synthetic network to the real data (red curve) is significant. As it can be observed the red curve (trained on synthetic, tuned on real data) does outperform the baselines on real data (green and orange curves). Please note that the blue curve has been evaluated on the synthetic data and therefore is not comparable to the other curves.

3.6.2.2 Retrieval based on RGB:

The retrieval results based on RGB values are shown in Fig. 3.16 left. Similar to optical flow based retrieval, the phenomena of synthetic data being helpful in retrieving real data is observed. However, the improvement margin is less significant. This is due to the higher accuracy of the network trained on real data (green).

3.6.3 Retrieving Synthesized Images:

As shown in Fig. 3.13, given an exocentric image I_{exo}, the synthesis network outputs a synthesized image I'_{ego}, and the corresponding ground-truth egocentric frame is called I_{ego}. In this experiment, we explore if the synthesis preserves higher level information. In other words, is I'_{ego} consistent

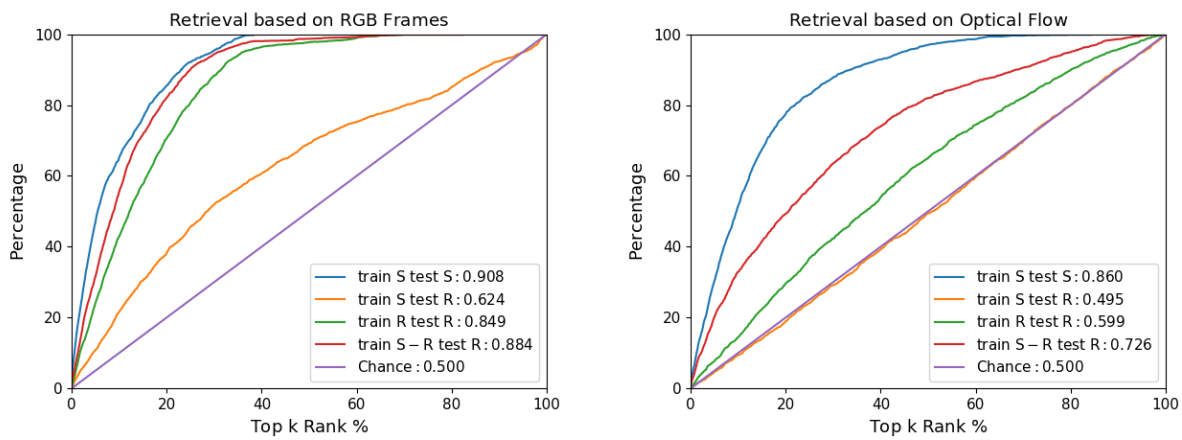

Figure 3.16: Retrieval performance based on RGB (left) and optical flow (right). S stands for synthetic data and R stands for real data.

Table 3.10: View Invariance-test based on Actions: In the synthetic dataset the chance level is 20% as there are 5 action classes. In the real dataset the chance level is 12% as there are 8 classes.

Retrieval Network \ View	**Egocentric View**	**Exocentric View**	**Both Views**
train Synthetic OF	37.71%	21.17%	27.33%
train Synthetic RGB	29.05%	27.29%	28.71%
trained Real OF	33.49%	28.18%	30.82%
trained Synthetic - Real OF	32.31%	32.97%	30.72%
trained Real RGB	42.58%	20.28%	24.16%
trained Synthetic - Real RGB	42.58%	20.43%	23.34%

with I_{ego} and I_{exo} in terms of high-level information? In order to answer this, we use the RGB retrieval network to extract egocentric features from the synthesized and ground truth egocentric images. In other words, we extract $f_{ego}(I'_{ego})$ and $f_{ego}(I_{ego})$ (where f_{ego} and f_{exo} are shown in Fig. 3.14.). We store all the features extracted from all synthesized egocentric images in F'_{ego}, the features from the ground-truth egocentric images in F_{ego}, and the features extracted from the exocentric images in F_{exo}. For each synthesized egocentric image in F'_{ego}, we retrieve its corre-

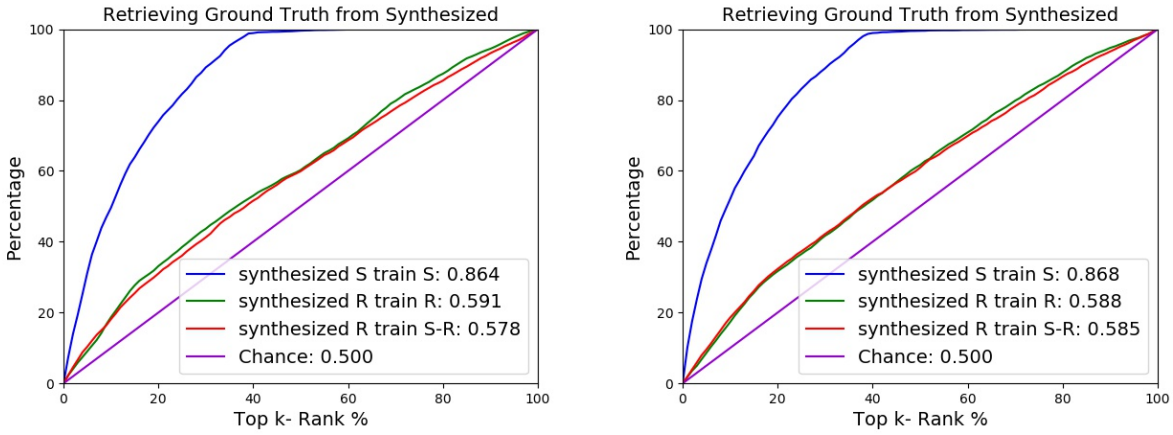

Figure 3.17: Retrieving the ground-truth egocentric, and exocentric images from the the synthesized images (left and right respectively). Similar to the figure 3.16, S stands for synthetic data and R stands for real data. Synthetic synthesized and ground truth images are fed to the retrieval network trained on synthetic data (blue). The real (synthesized and ground-truth) egocentric images are fed to the networks trained on real data (green: trained on real and red: trained on synthetic and fine-tuned on real).

sponding ground truth exocentric feature from F_{exo}. The retrieval results are shown in Fig. 3.17 (left). We also retrieve its corresponding ground truth egocentric feature from F_{ego}. The results are shown in Fig. 3.17. In both figures, the blue curve is the retrieval performance on the synthesized synthetic data, and the red and green curves show the retrieval on the synthesized real data using the different networks explained in the retrieval section.

3.6.4 View-invariance Test

Here we test the view-invariance of the retrieval network. To do so, we feed the training set (egocentric and exocentric RGB frames and optical flows) to our retrieval network and extract the features from their last fully connected layers (512 dimensions). In other words, we feed the egocentric frames to the ego stream and extract their features from the last fully connected layer.

We train two separate SVM classifiers on the features extracted from each view of the retrieval network: one SVM on egocentric features and action labels, and another on exocentric actions and labels. We then evaluate the performance of each of the SVMs (reported in Table 3.10 Egocentric view and exocentric view columns). A third SVM is then trained on all the features extracted from both views. In other words we pool all the features corresponding to each action independent of the fact that it is coming from the egocentric or exocentric stream. We then evaluate the performance of the third SVM on the first two. The classification performance of the SVM trained on both views does preserve the accuracy, and sometimes even outperforms the separately trained SVMs.

3.7 Discussion and Conclusion

In this chapter we introduce new synthetic and real datasets of simultaneously recorded egocentric and exocentric videos. We show that performing tasks such as retrieval and synthesis from third person to first person are possible. Future research can be done in the area of synthesis. Video generation is a possible extension of this effort. Also, embedding a view invariant representation in the bottleneck of the synthesis network can potentially unify the two tasks further. Other parameters such as camera and human pose and action labels can also be leveraged for better synthesis. Also, as we observed in our retrieval task, the synthetic data can be leveraged to address the lack of real data. We believe that this work provides useful datasets and baselines to address fundamental problems in relating first and third person images and videos.

CHAPTER FOUR: TASK ADAPTATION

Video Summarization via Actionness Ranking

Mohamed Elfeki and Ali Borji

Published in 2019 IEEE Winter Conference on Applications of Computer Vision (2019/1/7)

4.1 Abstract

To automatically produce a brief yet expressive summary of long videos, an automatic algorithm should start by resembling the human process of summary generation. Prior work proposed supervised and unsupervised algorithms that train models on learning the underlying behavior of humans by increasing modeling complexity or designing better heuristics to simulate human summary generation process. In this chapter, we take a different approach by analyzing a major cue that humans exploit for summary generation; the nature and intensity of actions. We empirically observed that a frame is more likely to be included in human-generated summaries if it contains a substantial amount of *deliberate* motion performed by an *agent*, which is referred to as *actionness*. Therefore, we hypothesize that learning to automatically generate summaries involves an implicit knowledge of actionness estimation and ranking. We validate our hypothesis by running a user study that explores the correlation between human-generated summaries and actionness ranks. To ensure reliable and consistent results, we run a consensus analysis between human subjects which exhibits a considerable agreement within obtained data. Based on the study findings that confirm our hypothesis, we develop a method for incorporating actionness data to explicitly regulate a learning algorithm that is trained for summary generation task. We assess the performance of our approach on four summarization benchmark datasets, and demonstrate an evident advantage compared to state-of-the-art summarization methods.

Figure 4.18: When generating summaries, humans often favor frames containing deliberate motion (such as a jumping man) over frames without **deliberate** motion (such as waterfall), even when natural/non-deliberate motion is more intense. The main question addressed here is whether we can gain insights from learning to recognize deliberate actions (i.e., actionness) to further assist video summarization.

4.2 Introduction

With the immense growth in the use of smart-phones and cameras, the amount of recorded visual data has become by far much more available than what can be attentively viewed. Each day 144,000 hours of video are uploaded to YouTube, which is almost 17 years worth of videos [55, 117, 41]. Moreover, recent statistics report that 245 million CCTV cameras are professionally installed around the world, actively surveying day-to-day activities [54]. Records in 2017 show that there are at least 2.32 billion active camera phones [118]. Estimates show that about 2.4 million GoPro body cameras were sold world-wide in 2015 [156]. This calls for efficient and automatic methods that quickly examine visual data and provide an informative briefing about the original videos. Video summarization addresses the problem of selecting a subset of video frames such that summary captures the most important and representative events of the original video.

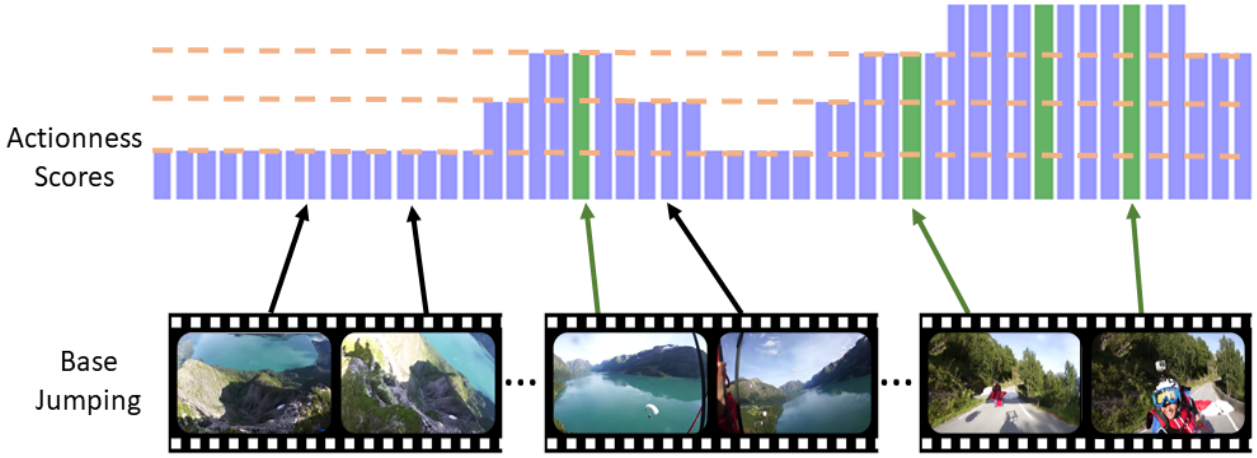

Figure 4.19: When examining human-generated summaries, we observe that they usually contain high degree of deliberate actions. In this chapter we put forth and examine the following hypothesis: *"Frames containing high magnitude of **deliberate** motion have a higher likelihood of being included within the video summary"*.

Several prior works made substantial efforts to better understand the video summarization problem and have proposed heuristic solutions (e.g., [83, 96, 12, 166, 100, 130]). The remarkable success of deep neural networks [72, 159, 111, 44] has motivated researchers in designing even more complex black-box models instead of a developing a profound understanding of the problem (e.g., [104, 181, 15, 64]). While increasing model complexity often helps in better modeling the latent patterns of data, it has the risk of overfitting to standard benchmark training video datasets and being sensitive to noise and irrelevant features, unless a proper learning objective is used. To address this challenge, here we seek to investigate a new learning objective that takes into account the role of deliberate actions performed by generic agents within the human-generated summaries and utilize this correlation to perform a robust automatic summarization. The premise of our work stems from our observation that humans tend to include frames with deliberate actions more frequently in the summary, since they tend to represent more "*unexpected and important*" events, and tell more about the story of the video.

Actions and motion patterns in videos present an intricate visual stimulation to the eyes of the viewer and thus become major cues when generating summaries for long videos. In the philosophy of actions [20], there are three aspects that define a generic action instance: i) it is carried out by an *agent*, ii) it requires an *intention*, and iii) it leads to *side-effects*. *Spatial Actionness* was introduced to quantify the likelihood of an image region to contain a generic action instance [16, 165]. Along the same lines, video summarization aims to localize temporal instances where important events occur. We propose to extend this definition to the temporal domain to better serve the summarization problem. That is, *Temporal Actionness* is the likelihood of a generic action to appear within a temporal video segment.

Temporal actionness ranking can assist an automatic summarization algorithm in localizing and quantifying the intensity of generic action instances. Consequently, it can also estimate the likelihood of including each event in the summary. Fig. 4.19 shows an example of a first-person video of a person performing base jumping. There are four distinct types of motion in this video: *running water, camera relative motion, a jumping partner, and first-person own-hand manipulation*; but only the last two instances qualify as strong temporal actionness which tend to constitute the vast majority of the summary.

Our main contributions in this paper is three-fold. First, we establish the concept of temporal actionness and study how it relates to video summarization. Second, we introduce a new set of actionness labels over four existing summarization benchmarks, and run a consensus and behavioral analysis on them to verify their consistency. Finally, we propose a method that utilizes temporal actionness to improve the summary generation through a multi-task learning formulation.

4.3 Related Work

In this section, we start by reviewing the concept of spatial actionness in the literature. Then, we briefly review Recurrent Neural Networks (RNN) and mention some of their applications in video processing. Finally, we conclude by discussing some prior approaches that have applied RNN models to the video summarization problem.

4.3.1 Actionness

The concept of spatial actionness was first introduced in [16] as the deliberate bodily movement performed by an agent; which is distinct from general instances of motion since it requires intention. They used Lattice Conditional Ordinal Random Fields to rank the regions of an image based on its likelihood of containing an action (i.e., ranking actionness).

Accurate and efficient ranking of spatial actionness was shown to benefit other related tasks [165, 173, 98, 85]. For example, Wang et al. [165] used a fully convolutional network to estimate spatial actionness. Then, they embedded the predicted actionness heat-map within a hybrid approach that performs action detection. Also, Ting et al. [173, 37] suggested a framework that performs action proposals by generating actionness curves via a snippet-level actionness classifier, then grouping them over time to produce the proposal candidates. Finally, Zhao et al. [184] proposed a temporal action proposal scheme called Temporal Actionness Tagging (TAG). This method uses an actionness classifier to evaluate the binary actionness probabilities for individual snippets. Our definition of temporal actionness is consistent with theirs, but also generalizes to agents other than humans as discussed in Section 3.1.

4.3.2 Recurrent Neural Networks (RNNs)

Since their introduction in [137, 167], RNNs have been commonly used to model sequential data. Unlike feed-forward networks (e.g., CNNs) whose output only depends on the input at the current time-step, RNN output also relies on previous time-steps. The basic formulation of RNN has the drawback of missing long-term dependencies due to the vanishing gradient problem [56]. Several extensions of RNNs have been introduced to resolve this problem. Popular approaches include: Long-Short Term Memory (LSTM) [57], and Gated Recurrent Unit (GRU) [18]. Both of these models have been successfully employed for applications such as video captioning using LSTM [162, 121, 172, 88], and action recognition and action proposals using GRU [11, 69, 164].

4.3.3 Video Summarization using RNNs

Because of their ability to process temporal data, RNNs have been widely used to train supervised and unsupervised video summarization models (e.g., [64, 104, 181, 15, 152, 182]). Zhang, et al. [181] were the first to use a supervised LSTM and a Multi-Layer Perceptron (MLP) while optimizing the Determinantal Point Process (DPP) maximum likelihood [79, 106, 77, 41]. DPP is used to quantify the diversity in the selected subset of frames which deems maximizing DPP to be equivalent to selecting a representative summary since the redundancy is minimized. Recently, Mahesseni et al. [104] presented an unsupervised video summarization framework by training an LSTM network in an adversarial manner to better model the complexity of the data. Further, Chen et al. [15] used a hybrid framework that utilizes GRU, MLP, and a temporal segmentation algorithm to perform the tasks of video summarization and video captioning simultaneously.

4.4 Relating Actionness to Summarization

In this chapter we hypothesize that human-generated summaries favor frames that contain deliberate motions over stationary or monotonous motions that are deemed boring. To test this hypothesis, we start by defining the type of motion that we expect to be a substantial component in human-generated summaries, which we refer to as temporal actionness. Then, we conduct a user study on human subjects investigating the relationship between temporal actionness and generated summaries. Finally, we conduct a consensus analysis on the obtained data to measure the agreement among subjects and a behavioral analysis to ensure the reliability of our findings.

4.4.1 Temporal Actionness

As discussed in Section 2.1, spatial actionness is defined as the likelihood of a certain region in an image to contain an action [16]. An image region is considered to contain an action based on the definition of actions in [20] as "what an agent can do with a deliberate bodily movement that leads to side-effects".

Our definition of actionness is consistent with the aforementioned definitions, but we extend it in two ways. First, we also consider non-human agents that perform deliberate motions, because human agents do not necessarily exist in the videos that are required to be summarized. For example, a swimming dolphin represents an action while a running river is not. Even though both of them contain similar magnitudes of motion but there is no intention in the latter.

Second, we adapt the actionness concept to the temporal domain, where we estimate the likelihood of a given video segment to contain an action. For biological agents, it is possible to predict the likelihood of the action from the agent's pose. However, since we are generalizing our definition to non-biological agents, their motion often is not distinguishable within a single frame. Thus, a video

segment is essential to determine the nature of motion. For instance, detecting a moving vehicle requires monitoring several frames to track the vehicle's location changes and to distinguish it from a stopped one.

We target a rank ordering of actionness rather than a binary classification of whether a segment contains an action (i.e., action proposal [11]) for two reasons. First, the fundamental notion of temporal actionness as "localizing when there is an action" immediately presents a difficulty: temporal segmentation remains a challenging and open problem. Some efficient methods exist for this purpose such as KTS [130], but the average f-score remains too low for robust use (about 0.41). Ranking makes it more plausible to provide a stratified quantification to the likelihood of a segment based on the prevalence of an action. Second, in any given video, often background actions (e.g., monotonous actions) are overlooked by the viewers as opposed to foreground abrupt actions. For instance, in a surveillance video, it is only natural to dismiss the background monotonous moving traffic, and monitor the abrupt motions around a building's entrance.

4.4.2 User Study

To estimate actionness, we first used KTS algorithm [130] to produce semantically consistent variable-size segments that contain atomic semantic meanings. Then, for each segment, we asked five users to label it by selecting the appropriate rank from the following scales:

0: No action (No deliberate motion by an agent)

1: Background action (Weak indication of an action)

2: Partial foreground action (Strong action indication covering a minor part of the segment)

3: Active foreground action (Strong action indication covering a major part of the segment)

For a tractable annotation process, we subsampled the videos to 1 fps. Then, we constructed the

displayed segment to contain all the frames in a grid display allowing the users to see all the frames of one segment simultaneously. Before starting the process, users underwent a training stage to understand the task and the procedure. They were asked to rank actionness on four videos. After training, the users were asked to perform the same task on four benchmark summarization datasets: SumMe [50], TVSum [148], Youtube [21], and OVP [38]. Videos used during the user training stage were discarded in model development.

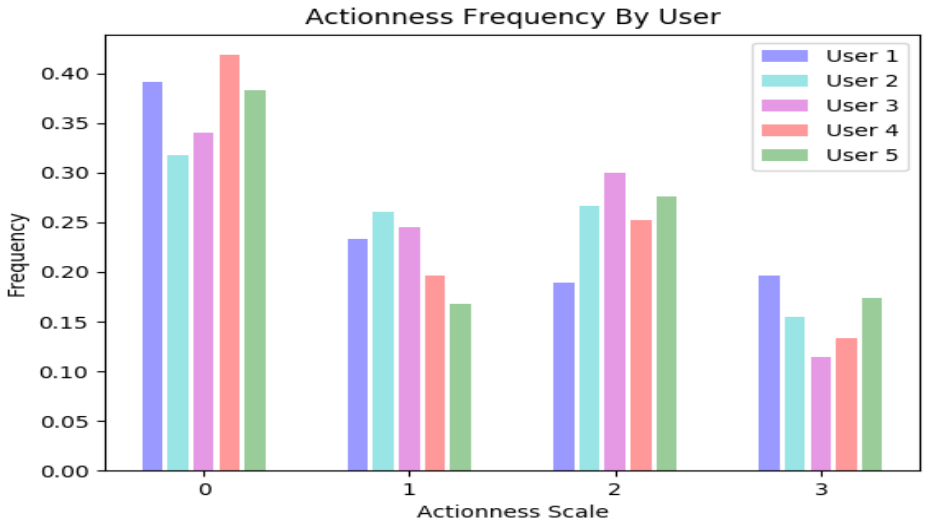

Figure 4.20: *How often each user chose a given actionness scale in the annotations?* Having close frequencies indicates a general agreement between the users.

4.4.3 Data Analysis

4.4.3.1 Consensus analysis

To ensure the validity of the annotations, we measured the consensus among users using two metrics. The first metric is the f-1 score. We computed the average pairwise f1-measure to estimate the agreement among the annotators for each scale. We obtained 0.55, 0.40, 0.48, and 0.51 for SumMe,

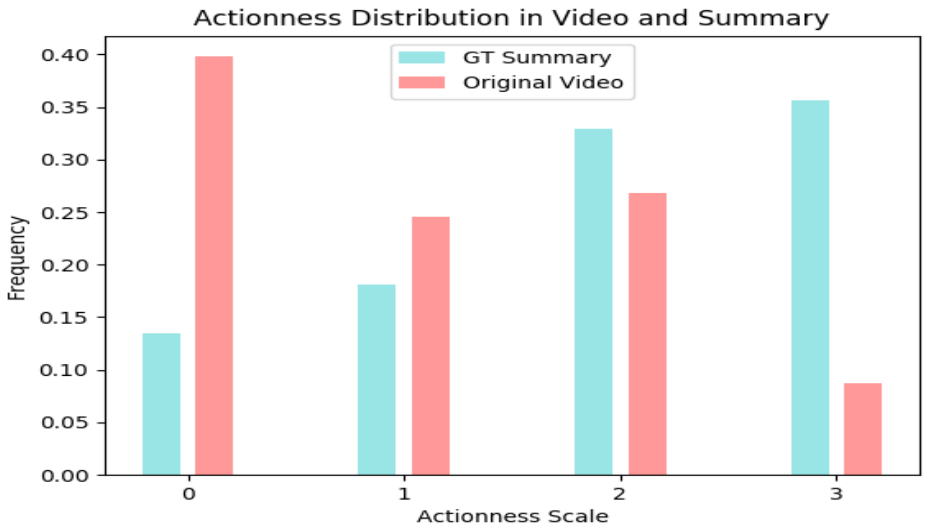

Figure 4.21: *Do GT summaries contain high actionness?* GT summaries mostly consist of scale-three actionness, while original videos mostly contain scale-zero actionness.

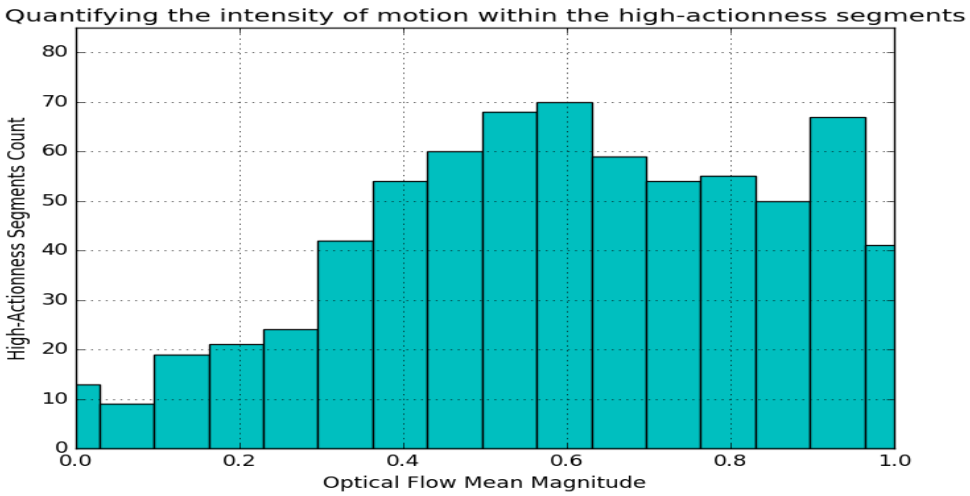

Figure 4.22: *Were the annotators just looking for abrupt motions?* Non-abrupt motions also exists vastly in the selected summaries, mostly with optical flow changes $\geq 25\%$

TVSum, OVP, and Youtube datasets, respectively. The second metric is the rank-frequency over original videos for each user. That is, how often each user chose a given scale for all the videos of

the annotation? Fig. 4.20 shows the frequency ranks for all users. We observe that ratios by users are close to each other for all the scales, which along with the f-1 scores demonstrates evident consensus among users.

4.4.3.2 Do summaries contain high actionness?

To answer this question, we computed the average frequency of each actionness scale in both of the ground-truth summary and the original video. Fig. 4.21 demonstrates that scale-three actionness frames seem to be the dominant majority rank among the summary despite their minority existence in the original video. *Hence, frames containing high actionness are more likely to be included in the summary*.

4.4.3.3 Were the annotators just looking for abrupt motions?

For a more extensive verification, we examine if the users tended to choose segments containing abrupt motion (i.e., high magnitudes of motion) as representation for the high-actionness segments. To answer this question, we first need to provide an evaluation for abrupt motion. We calculated the mean magnitude of optical flow for each of the segments, and normalized it across each video. Then, we computed the histogram plot of the segments scored by the users as level-three actionness sorted by their normalized mean magnitude of optical flow. As shown in Fig. 4.22, the selected segments are distributed among a wide variation of optical-flow intensities. This shows that users were not merely selecting the most abrupt motion segments as representatives for the deliberate actions required in high actionness.

4.4.3.4 Oracle labels

Having established our hypothesis, we seek to utilize the data obtained from the study to further improve the automatic video summarization algorithms. In order to train a supervised learning model, we need to produce a single set of labels out of multiple annotations for each video. This is often referred to as *Oracle Labels* set. We follow the algorithm proposed in [41, 77] that greedily selects the segment that results in the largest marginal gain on the f-1 score computed between the users' annotations. To produce frame-level labels, we consider all the frames within a segment to have its ranking label.

4.5 Approach

In this section we propose a model that incorporates actionness ranking task to regularize video summarization.

4.5.1 Overview

Figure 4.23 shows an overview of our framework. The input is a video of n frames. First, a visual encoder ϕ (i.e., a pretrained CNN) is used to extract spatial features for each frame. Next, the extracted features are sent to a sequential encoder (i.e., a Bi-directional GRU) to extract their corresponding temporal features. GRU is used as a sequential encoder because it has fewer parameters than LSTM, which results in faster training and a less risk of overfitting, and shown to perform on par to the LSTM [18]. Next, we aggregate both types of the features, spatial and temporal, to generate a comprehensive spatio-temporal feature vector for each frame. These features represent the visual information of the current frame as well as encode all the temporal information from other

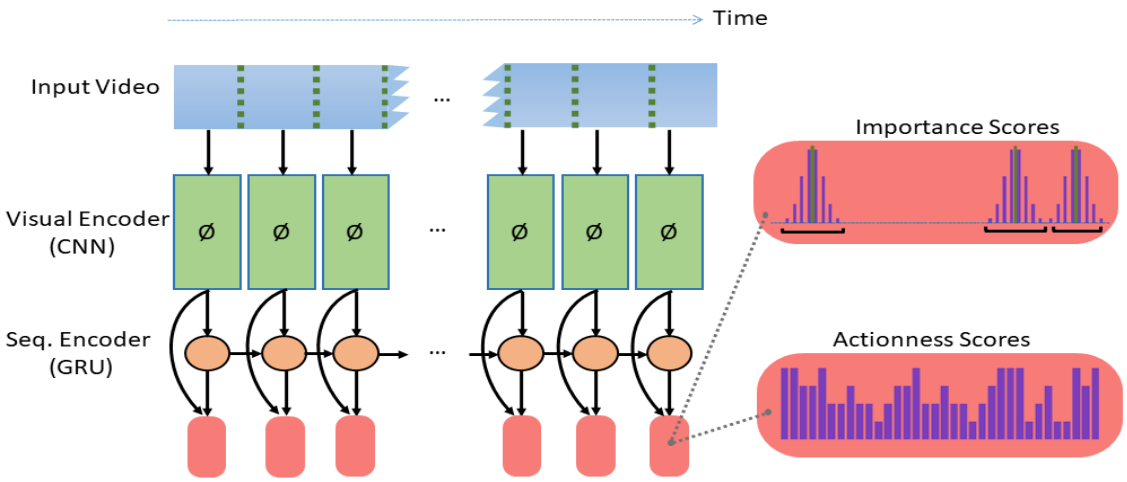

Figure 4.23: Using actionness ranking (i.e., actionness level classification of each frame) to regularize the learning of video summarization. Summarization is learned by maximizing diversity within the selected subset. Training the framework in a multi-task learning fashion with an accessory task of actionness ranking, improves the learning of the main task (i.e., video summarization).

frames in the video. Finally, the aggregate features are mapped to the actionness and importance scores using two independent MLPs.

The framework is trained to learn two tasks: 1) summarization by minimizing importance estimation loss, and 2) actionness ranking by minimizing actionness classification loss. The framework is optimized by applying a regularized multi-task learning paradigm [29]. Imposing a regularization term in a joint loss is aimed to penalize the unnecessary complexity of the original learning problem that might cause overfitting to training data, while enforcing learning task relationship.

By combining the two losses into a single joint loss, the network is trained to learn a set of trainable parameters θ such that:

$$argmin_{\theta} \; S(\theta) + \lambda R(\theta), \tag{4.14}$$

where $S(\theta)$ is the summarization loss (section 4.2), $R(\theta)$ is the actionness classification loss (sec-

tion 4.3) which acts as a regularizer, and λ is the regularization weight used to force both the losses to operate on comparable ranges, preventing the learning to be biased towards one of the losses.

4.5.2 Importance Estimation

Importance scores (i.e., summarization labels) are binary labels that indicate the frames selected to be a part of the summary: 1 for selected frames, and 0 otherwise. The problem with this type of labeling is that frames within the same segment tend to have similar semantic features, therefore the annotators could have chosen any other frame within a selected frame's segment (i.e., key segment). To reduce the effect of the inherent noise in the labels, we apply Gaussian smoothing as a preprocessing step. Particularly, binary labels are converted to real-values where the mean is the selected frame within the summary, and the Gaussian distribution is sampled across its key segment (see Fig. 4.23). Thus, the framework would not be penalized for choosing a frame within a key segment as much as it would be penalized for choosing a frame outside a key segment.

Increasing the diversity within the selected subset is equivalent to choosing a representative subset since the redundancy is minimal. Following [41, 10], we follow the decomposition in [77] to compute the marginal kernel L_y as a of a Gram matrix in the following manner:

$$L_{ij} = q_i \phi_i^\top \phi_j q_j; \ \forall i, j \in y \tag{4.15}$$

where ϕ_i can be seen as a representative feature vector, and q_i is quality score of frame i in the selected subset y. Similar to [181], we construct the features with a dimensionality of 256 for each frame, and the quality score as a single scalar for every frame. In our framework, we apply two independent MLPs with the aforementioned dimensions to obtain ϕ and q and compute the marginal DPP kerenl as in Eq. 5.18.

Finally, we optimize the Maximum Likelihood Estimation (MLE) of the normalized marginal DPP kernel that quantifies the diversity in the ground-truth summaries y as follows:

$$S(\theta) = -log\left[\frac{det(L_y)}{det(L+I)}\right] \tag{4.16}$$

where L is the marginal kernel of the ground-set of all the frames in the video, and I is the identity matrix.

4.5.3 Actionness Ranking

This task aims to provide a regularization term to the joint loss (Eq. 4.14) which is determined by classifying the actionness scale v of each frame; $v \in \{0, 1, 2, 3\}$. We train an independent MLP to map the spatio-temporal features of each frame to an actionness rank using the categorical cross entropy loss as follows:

$$R(\theta) = -\sum_{i=1}^{n}\sum_{j=0}^{3} t_{i,j} log(p_{i,j}), \tag{4.17}$$

where $p_{i,j}, t_{i,j}$ are the predicted and target values of actionness rank j for the i-th frame.

4.6 Experiments

In above sections, we proposed that deliberate motion provides a significant cue when humans are summarizing a given video. Then, we established this hypothesis by performing a user study among multiple human subjects that were asked to rank the magnitude of deliberate motion. By analyzing the study results, it is clear that a significant portion of the summary includes high intensity of deliberate motion, as opposed to the original video contents. Therefore, we introduced

an approach that can rank the intensity of deliberate motion and uses this knowledge to improve the performance to perform a better video summarization. In this section, we run an extensive set of experiments where we show the effect of learning the actionness in learning summarization.

4.6.1 Datasets

We evaluated our approach on four summarization benchmark datasets: SumMe [50], TVSum [148], Open Video Project (OVP) [38], and Youtube [21]. The first dataset consists of 25 user videos covering multiple events such as bears climbing a tree and cooking. It contains both first-person and third-person videos with lengths varying from 1.5 to 6.5 minutes. The second dataset consists of 50 Youtube videos from 10 categories of the TRECVid Multimedia Event Detection (MED), 5 videos per category. They vary in length from 1 to 5 minutes and include both first and third person videos.

The third and fourth datasets are quite large. We use the same subset of videos used in [21, 104, 181], 50 videos from OVP, and 39 videos from Youtube. OVP videos contain mostly news reports and documentary clips that vary in length from 1 to 4 minutes. All of them are third-person videos. The last dataset contains news and sports videos (third-person videos) with lengths varying from 1 to 10 minutes.

4.6.2 Experimental Setup

For a fair comparison with the related approaches, we evaluate our method using the keyshot-based metric similar to [181, 104]. We first convert frame-level scores to shot scores by applying the KTS algorithm [130] that generates semantic shots. The resulting shots are ranked based on their importance score, which is the average score of the frames in that shot. By applying the Knapsack

Table 4.11: F1-scores for several test configurations. Canonical: Train on 80% of a dataset, test on the remaining 20%. Augmented: Train on one dataset, test on the other. Transfer: Train on one dataset + OVP + YouTube, test on the other.

Model	Canonical		Augmented		Transfer	
	SumMe	TVSum	SumMe	TVSum	SumMe	TVSum
[29]	26.6	-	-	-	-	-
[15]	39.7	-	39.7	-	-	-
[14]	39.5	-	39.3	-	-	-
[55]	40.9	-	40.9	-	38.5	-
[56]-vsLSTM	37.6	54.2	37.6	54.2	41.6	57.9
[56]-dppLSTM	38.6	54.7	38.6	54.7	42.9	59.6
[33]-DPP	-	-	39.1	51.7	43.4	59.5
[33]-Sup	-	-	41.7	56.3	43.6	**61.2**
Ours-Basic	37.9	54.6	38.8	54.8	43.1	59.6
Ours-FT	38.7	54.9	42.3	56.1	43.8	59.3
Ours-Reg	**40.1**	**56.3**	**45.8**	**59.1**	**46.1**	60.1

algorithm, a subset of the highest ranked keyshots are selected such that the total duration of the generated summary is less than 15% of the original video. We report the average f1-scores to evaluate the predicted summary as compared to the ground-truth summary.

4.6.2.1 Implementation Details

Similar to [104, 181], we use the output of the pool5 layer of GoogLeNet [158] architecture trained on ImageNet [22] as the visual encoder for our framework to extract a 1024 dimension spatial feature vector for each frame. Then, we use a single-layer GRU with 256 hidden units as the sequential encoder and 256 hidden units MLPs for both of the optimization tasks. Similar to the training setup of [181], we run our model for 100 iterations in the training stage and stop the training if the validation f1-score does not improve for more than 5 consecutive iterations. The validation split is set to be 20% random subset of the training data. We use Adam optimizer to train our framework with learning rate of 0.001. To learn the task of actionness ranking, we set λ

to 0.003. The value of λ was selected to make both of the losses operate on close ranges so that none of them bias the optimization while training the network.

4.6.3 System Performance

4.6.3.1 Test Configurations

We follow [181, 104] to evaluate our method in three test configurations. In the first configuration (Canonical), we use 80% of one dataset to train the method, and test the method on the remaining 20% of the same dataset. In the second configuration (Augmented), TVSum and SumMe datasets are used together - one dataset is used to train the method while being tested on the entire other dataset. In the last configuration (Transfer), we adapt the same paradigm as the second configuration but augment the training set with OVP and Youtube datasets, which improves the results on SumMe and TVSum.

4.6.3.2 Baselines

We conduct an extensive comparison with the state of the art methods [50, 51, 180], two models from [181]: LSTM+MLP (vsLSTM) and LSTM+MLP+DPP (dppLSTM), and two models from [104]: Unsupervised DPP (DPP) and supervised model (SUP).

Also, to perform an ablation study on our model, we introduce three variants of our approach. First, *Ours-Basic* is our model without the actionness regularization;. It reduces the model's complexity to be close to [181], however, our model uses GRU instead of LSTM and performs Gaussian smoothing preprocessing on the labels. Second, *Ours-FT* is the same as the basic model, but the sequential encoder is first trained for human-based action localization, then the entire framework

is fine-tuned for video summarization. To train the GRU for action localization, we follow [114] to train the sequential encoder on GoogLeNet features for action recognition task on UCF-101 [149] for 100 epochs, then fine-tune it for action localization on THUMOS-14 [65] for another 100 epochs. The last model is *Ours-Reg*, which is a model that is trained for simultaneous video summarization and actionness estimation as discussed in Section 4.

4.6.3.3 Summarization Evaluation

Table 4.11 shows the f-1 scores of our models compared to the state-of-the-art methods. As shown, Ours-Basic performs similarly to vsLSTM and dppLSTM. Training our model on the action recognition labels prior to summarization (Ours-FT) performs on par with the state-of-the-art methods. However, the model that is trained for actionness estimation, that is considering deliberate motions performed by generic agents (not just humans unlike Ours-FT), significantly outperforms all other methods in most of the settings (Ours-Reg).

Table 4.12: Actionness Classification Accuracy of Ours-Reg: In all the settings our model learned to estimate actionness better than the chance level.

	SumMe	TVSum
Chance	*36.6*	*28.1*
Canonical	39.7	30.3
Augmented	42.8	32.6
Transfer	41.8	29.5

4.6.3.4 Actionness Evaluation

To investigate whether actionness helps summarization, we ran two analyses. First, we verify that our model effectively learns the actionness ranking task by computing the actionness classification

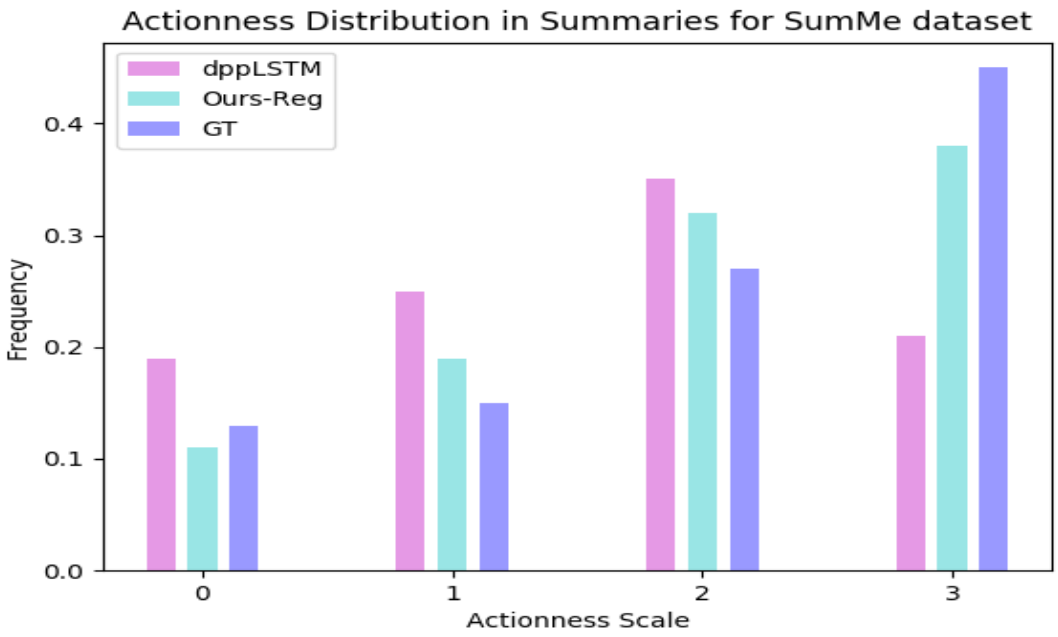

Figure 4.24: Distribution of actionness scales over summaries of SumMe dataset. Our model better resembles the GT than dpp-LSTM [181].

accuracy in all test configurations. As shown in Table 4.12), Ours-Reg performs significantly better than chance, indicating that the model actually learns actionness estimation and does not dismiss it from the learning procedure. Second, we compute the distribution of actionness scales in the ground-truth summary, Ours-Reg, and [181] over the SumMe dataset for test configuration 1. As shown in Fig. 4.24, our model resembles the ground-truth summary better than [181]. The two results suggest that learning actionness ranking is indeed useful for better video summarization.

4.7 Conclusion

In this chapter, we present a further step in analyzing and understanding the video summarization problem. We hypothesize that humans actively rely on **deliberate** motion and action cues -among

other cues- to generate a brief summary that best expresses long videos. To test this hypothesis, we run a user study to investigate the correlation between human-generated summaries and actionness ranking. Further, we conduct a consensus analysis on the data obtained from the users to ensure the data reliability and agreement among the users. The findings of the study show a substantial likelihood of including frames containing high actionness ranks within the summaries.

Therefore, we propose a new method that uses the actionness information to better learn the task of video summarization. We use a recurrent neural network that is trained for video summarization while being explicitly regularized to learn the actionness ranking task in a multi-task learning formulation. The evaluation on four benchmark summarization datasets shows a distinct improvement by our approach over several state-of-the-art summarization methods.

CHAPTER FIVE: OBJECTIVE GENERALIZATION

Multi-stream dynamic video Summarization

Mohamed Elfeki, Aidean Sharghi, Srikrishna Karanam, Ziyan Wu, and Ali Borji

Under review in 2021 IEEE Winter Conference on Applications of Computer Vision (2021/1/7)

5.1 Abstract

With vast amounts of video content being uploaded to the Internet every minute, video summarization becomes critical for efficient browsing, searching, and indexing of visual content. Nonetheless, the spread of social and egocentric cameras creates an abundance of sparse scenarios captured by several devices, and ultimately required to be jointly summarized. In this paper, we discuss the problem of summarizing videos recorded independently by several dynamic cameras that intermittently share the field of view. We present a robust framework that (a) identifies a diverse set of important events among moving cameras that often are not capturing the same scene, and (b) selects the most representative view(s) at each event to be included in a universal summary. Due to the lack of an applicable alternative, we collected a new multi-view egocentric dataset, Multi-Ego. Our dataset is recorded simultaneously by three cameras, covering a wide variety of real-life scenarios. The footage is annotated by multiple individuals under various summarization configurations, with a consensus analysis ensuring a reliable ground truth. We conduct extensive experiments on the compiled dataset in addition to three other standard benchmarks that show the robustness and the advantage of our approach in both supervised and unsupervised settings. Additionally, we show that our approach learns collectively from data of varied number-of-views and orthogonal to other summarization methods, deeming it scalable and generic.

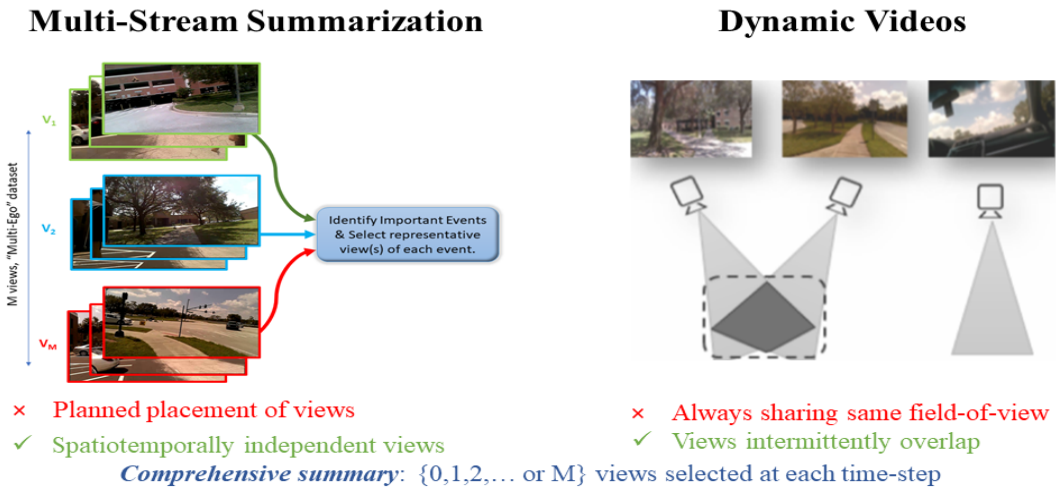

Figure 5.25: Several views are recorded independently and intermittently overlap their fields-of-view. Our approach dynamically accounts for inter- and intra-view dependencies, providing a comprehensive summary of all views.

5.2 Introduction

In a world where nearly everyone has several mobile cameras ranging from smart-phones to body-cameras, brevity becomes no longer an accessory. It is rather essential to efficiently extract relevant contents from this immense array of static and moving cameras. Video summarization aims at selecting a set of frames from a visual sequence that contains the most important and representative events. Not only is summarization useful for efficiently extracting the data substance, it also serves many other applications such as video indexing [59], video retrieval [169], and anomaly detection [35].

We consider a generic setting where multiple users record egocentric footage that is both spatially and temporally independent. Users are allowed to move freely in an uncontrolled environment. As such, cameras' fields-of-view may or may not overlap through the sequence. Unlike fixed-camera videos, egocentric footage often displays rapid changes in illumination, unpredictable camera mo-

tion, unusual composition and viewpoints, and often complex hand-object manipulations. Thus, a universal summary should capture a diverse set of events across all different viewpoints, while being robust to egocentric noise. Additionally, whenever an event is being captured by more than one camera, the summary should only include the most representative view and dismiss the rest. However, if some views(streams) are complementary or mutually exclusive and all represent important events simultaneously, all need to be included in the summary.

This setting presents itself in several real-life scenarios where many egocentric videos are required to be summarized collectively. For instance, rising claims of police misconduct led to a proliferation of body cameras recordings [154, 7]. Typical police patrols contain multiple officers working 10-12 hour shifts. Although it is crucial to thoroughly inspect key details, manually going through 10-hour video content is extremely challenging and prone to human errors. Multiplying shift lengths by the number of officers on duty, it is obvious that there are copious amounts of data to analyze with no guiding index. A similar example occurs at social events such as concerts, music shows, and sports games. Those events tend to be recorded by many several cameras simultaneously that are dynamically changing their fields-of-view. Nevertheless, the final highlight summary of such events is likely to contain frames from all cameras.

Despite considerable progress in single-view video summarization for both egocentric and fixed cameras (e.g., [181, 131, 40, 103]), those techniques are not readily applicable to summarizing multi-view videos. Single-view summarizers ignore the temporal order by processing simultaneously-recorded views in a sequential order to fit as a single-view input. This results in redundant and repetitive summaries that do not exhibit the multi-stream nature of the footage. On the other end of spectrum, the literature of multi-view video summarization mainly focuses on fixed surveillance camera summarization (e.g., [124, 125, 123]). This enables some methods to rely on geometric alignment of cameras inferring the relationship between their fields-of-view and utilizing it for a representative summary (e.g., [6, 36]). Thus, previous work mostly uses unsupervised methods

that are based on heuristic-based objective functions, which are not suitable to a dynamic change in cameras' geometric positioning. A key motivation for our work is to generalize the multi-stream summarization to accommodate dynamic cameras and extend the capacity of existing supervised and unsupervised summarization techniques.

5.2.1 Contributions

We extend single-view and fixed-cameras methods to be applied on the generalized multi-stream dynamic-cameras setting. We propose a novel adaptation of the widely used Determinantal Point Process (DPP) [181, 103, 40, 142], Multi-DPP, generalizes it to accommodate multi-stream setting while maintaining the temporal order. Our approach is orthogonal to other summarization approaches and can be embedded with fixed- or moving-cameras and operating on a supervised or unsupervised setting. Furthermore, our method is shown to be scalable (can be trained on labels of any available number-of-views in the supervised setting) and generic (encompasses both single-view and fixed-cameras settings as special cases). Since no existing dataset is readily applicable to evaluate such setting, we collect and annotate a new dataset, Multi-Ego. With extensive experiments, we show that our method outperforms state-of-the-art supervised and unsupervised baselines on our generic configuration as well as the special case of fixed-cameras multi-view summarization.

5.3 Related Work

5.3.1 Single-View Video Summarization

Among many approaches proposed for summarizing single-view videos supervised approaches usually stood out with best performances. In such a setting, the purpose is to simulate the pat-

terns that people exhibit when performing the summarization task, by using human-annotated summaries. There are two-factor influence the supervised models' performance: (a) reliability of annotations, and (b) framework's modeling capability. Ensuring the reliability of annotations is evaluated based on a consensus analysis as in several benchmark datasets [101, 148, 82]. As for the modeling capabilities, supervised approaches vary in their modeling complexity and effectiveness [40, 52, 178, 49, 175, 31].

Recently, [139] proposed to use convolutional sequences to summarize videos in both supervised and unsupervised settings. By formulating the problem as a sequence labeling problem, they established a connection between semantic segmentation and video summarization and used networks trained on the former to improve the latter. Others have formulated the summarization problem within a reinforcement learning paradigm either with an explicit classification reward as in [186] or a more subtle diversity-representativeness reward [187]. Both approaches provided relatively competitive results on single-view, nonetheless they suffer from unstable training in the multi-view setting as we detail in the experiments section.

Recurrent Neural Networks in general, and Long Short-Term Memory (LSTM) [57] in particular has been widely used in video processing to obtain the temporal features in videos [162, 121, 189, 93]. In the recent years, using LSTMs has been a common practice to solve video summarization problem [64, 153, 170, 183, 171, 90, 27]. For example, Zhang et al. [181] use a mixture of Bi-directional LSTMs (Bi-LSTM) and Multi-Layer Perceptron to summarize single-view videos in a supervised manner. They maximize the likelihood of Determinantal point processes (DPP) measure[80, 46, 163] to enforce diversity within the selected summary. Also, Mahasseni et al. [103] present a framework that adversarially trains LSTMs, where the discriminator is used to learn a discrete similarity measure for training the recurrent encoder/decoder and the frame selector LSTMs.

5.3.2 Multi-view Video Summarization

Most multi-view summarization methods tend to rely on feature selection in an unsupervised optimization paradigms [120, 123, 125, 124, 138, 109]. Fu et al. [36] introduce the problem of multi-view video summarization as tailored for fixed surveillance cameras. They construct a spatiotemporal graph and formulate the problem as a graph-labeling task. Similarly, in [124, 123, 109] authors assume that cameras in a surveillance camera network have a considerable overlap in their fields-of-view. Therefore they apply well-crafted objective functions that learn an embedding space and jointly optimize for a succinct representative summary. Since those approaches target fixed surveillance cameras, they rightfully assume a significant correlation among the frames along the same view over time. In our generalized setting, cameras move dynamically and contain rapid changes in the field-of-view rendering the aforementioned assumption weak and make the problem harder to solve.

A similar problem was introduced in literature by Arev et al. [6], entailing editing footage recorded from social cameras. They propose a graph-based approach that provides an automatically generated cut of a specific length out of the videos from all users. And by constructing the 3D structure from motion, they obtain a universal knowledge of the event. While their technique may work in certain scenarios, constructing 3D structure is unattainable in most situations especially if cameras are dynamically moving and containing considerable egocentric noise.

5.4 Dataset

While a number of multi-view datasets exist (e.g. [36, 120]), none of them are recorded in egocentric perspective. Therefore, we collect our own data that aligns with the established problem setting. We asked three users to independently collect a total of 12 hours of egocentric videos

while performing different real-life activities. Data covers various uncontrolled environments and activities. We also ensured to present different levels of interactions among the individuals: (a) two views interacting while the third one is independent, (b) all views interacting with each other, and (c) all views independent of each other. Then, we extracted 41 different sequences that vary in length from three to seven minutes. Each sequence contains three views covering a variety of indoors and outdoors activities. We made the data more accessible for training and evaluation by grouping the sequences into 6 different collections.

To put our dataset size (41 videos of 3-7 minutes) in perspective, we refer to the most commonly used summarization benchmarks: SumMe (25 videos of 2-4 minutes), TVSum (50 videos of 2-4 minutes) [148], Office (4 videos of 11 minutes), Lobby (3 videos of 8 minutes) and Campus (4 videos of 15 minutes) [36, 120]. Even though that collecting larger sizes and longer videos is desirable, nonetheless, annotating simultaneously collected views by several annotators is a notoriously hard task. In the following section, we shed some light on the difficulties encountered in that task and we propose *annotating-in-stages* approach to reduce the annotation uncertainty. More details about data-collection and a behavioral analysis on the obtained annotations are provided in Appendix B.5-7.

5.4.1 Collecting User Annotations

To annotate and process the data for the summarization task, we sub-sample the videos uniformly to one fps following [142]. Then, every three consecutive frames are combined to construct a shot for an easier display to annotators. The number of frames per shot was chosen empirically to maintain a consistent activity within one shot.

We asked five human annotators to perform a three-stage annotation task. In *stage one*, they were asked to choose the most interesting and informative shots that represent each view independently

without any consideration towards the other views. To construct two-view summaries in *stage two*, we only displayed the first two views simultaneously, while asking the users to select the shots from any of the two views that best represent both cameras. Similar to stage two, in *stage three* the users were asked to select shots from any of the three views that best represent all the cameras. It is worth noting that the annotators were not limited to choose only one view of a certain shot, and they could choose as many as they deem important.

The *annotating-in-stages* procedure explained above was employed due to the human's limited capability in keeping track of unfolding storylines along multiple views simultaneously. Consequently, using this technique resulted in a significant improvement in the consensus between user summaries compared to when we initially collected summaries in an unordered annotation task.

5.4.2 Analyzing User Annotations

To ensure the reliability and consistency of the obtained annotations, we perform a consensus analysis using two metrics: average pairwise f1-measure and selection ratio. Following [148, 142, 131], we compute the average pairwise f1-measure to estimate the frame-level overlap and agreement. We calculated the f1-measure for all possible pairs of users' annotations and averaged the results across all the pairs, obtaining an average of 0.803, 0.762, and 0.834 for the first, second, and third stage respectively.

5.4.3 Creating Oracle Summaries

Finally, training a supervised method usually requires a single set of labels. That means in our case, we need to use only one summary per video, which is often referred to as *Oracle Summary*. To create an oracle summary using multiple human-created summaries, we follow [40, 76] to

greedily choose the shot that results in the largest marginal gain on the f-score, and iteratively keep repeating the greedy selection until the length of the summary reaches 15% of the single-view length.

5.5 Approach

We first discuss the original single-view DPP criterion in Section 4.1. Then, we illustrate how we adapted the formulation to the Multi-stream setting and the generalized supervised and unsupervised formulation in Section 4.2. In section 4.3, we detail the design of our summarization approach and we conclude by analyzing the scalability of our supervised system in Sec 4.4.

5.5.1 Determinantal Point Process (DPP)

DPP is a probabilistic measure that provides a tractable and efficient means to capture negative correlation with respect to a similarity measure [102, 80]. Formally, a discrete point process $\mathcal{P}$ on a ground set $\mathcal{Y}$ is a probability measure on the power set 2^N, where $N = |\mathcal{Y}|$ is the ground set size. A point process $\mathcal{P}$ is called determinantal if $\mathcal{P}(y \subseteq Y) \propto \det(L_y); \forall\, y \subseteq Y$. Y is the selection random variable sampled according to $\mathcal{P}$ and L is a symmetric semi-definite positive matrix representing the kernel.

Kulesza et al. [75] proposed modeling the marginal kernel L as a Gram matrix in the following manner:

$$\mathcal{P}(y = Y) \propto \det(\Phi_y^\top \Phi_y) \prod_{i \in y} q_i^2, \tag{5.18}$$

When optimizing the DPP kernel, this decomposition learns a "quality score" of each item, where $q_i \geq 0$. It also allows learning a feature vector Φ_y of subset $y \subseteq \mathcal{Y}$. In this case, the dot product $\Phi_y = [\phi_i|...|\phi_j]$, where $\phi_i^\top \phi_j \in [-1, 1]; \forall i, j \in y$ is evaluated as a "pair-wise similarity measure" between the features of item i, ϕ_i and the features of item j, ϕ_j. Thus, the DPP marginal kernel L_y can be used to quantify the diversity within any subset y selected from a ground set $\mathcal{Y}$. Choosing a diverse subset is equivalent to a brief representative subset since the redundancy is being minimized. Hence, it is only natural that a considerable number of document and video summarization approaches use this measure to extract representative summaries of documents and videos [76, 103, 40, 163].

5.5.2 Adapting DPP to Multi-stream: Multi-DPP

The standard DPP process described above is suitable for selecting a diverse subset from a single ground set. However, when presented with several temporally-aligned ground sets $\{\mathcal{Y}_1, \mathcal{Y}_2, ..., \mathcal{Y}_M\}$, the standard process can only be applied in one of two settings: either (a) merging all the ground sets into a single ground set $\mathcal{Y}^{merge} = \{\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup ... \cup \mathcal{Y}_M\}$ and selecting a diverse subset out of the merged ground set, or (b) selecting a diverse subset from each ground set and then merging all the selected subsets $Y^{merge} = \{Y_1 \cup Y_2 \cup ... \cup Y_M\}$.

Even though that the former setting preserves the information of all elements of the ground sets, but it causes the complexity of the subset selection to exponentially grow. In practice, this leads to an accumulation of error due to overflow and underflow computations as well as substantially slower running-time. Additionally, latter setting assumes no-intersection between features of the different ground-sets. This is essentially inapplicable if the ground-sets have a significant dynamic feature overlap, leading to redundancy and compromising the very purpose of the DPP. To address these shortcomings, we propose a new adaptation of Eq. 1, called *Multi-DPP*.

In Multi-DPP, ground sets are processed in parallel allowing any potential feature overlap across the ground sets to be processed temporally-appropriate and keeping a linear growth with respect to the number of streams. For every element in the ground sets, we need to represent two joint quantities: features and quality, such that they follow the following four characteristics. First, we need a model that can operate on any number of streams (i.e., generic to any number of ground sets M). Second, we need a joint representation of the features at each index, such that it only selects the most effective ones (i.e., invariance to noise and non-important features). Third, we need a joint representation of the qualities at each index, such that is affected by the quality of each ground set at a particular index (i.e., variance to the quality of each ground set). Forth, we need to ensure that our adaptation follows the DPP decomposition in Eq. 5.18, by selecting joint features $\phi_i^\top \phi_j \in [-1, 1]$, and joint qualities $q_i \geq 0; \forall i, j \in y$.

To account for joint features, we apply max-pooling choosing the most effective features across all ground sets at every index, which satisfies the feature decomposition in Eq. 5.18. Selecting joint qualities -on the other hand- needs to account for the quality of each ground set in every index. We use the product of all the qualities at each index. This deems the joint quality at each index to be dependent on all ground-sets while also ensuring $q^m \leq 1$. Therefore, we generalize the Determinantal Point Process based on the decomposition in Eq. 5.18 as follows:

$$\begin{aligned} \mathcal{P}(Y = y) &\propto \det(\Phi_y^\top \Phi_y) \prod_{m=1}^{M} \prod_{i \in y_m} [q_i^m]^2 \\ \phi_j &= max(\phi_j^1, ..., \phi_j^M) \; ; \forall j \in y \end{aligned} \tag{5.19}$$

where M is the number of the ground sets and y_m is the subset selected from ground set m. This decomposition allows both a scalable multi-stream (by constructing a joint feature representation with max-pooling), and monitoring the egocentric-introduced noise (by learning an independent

quality measure for each view at each time-step).

5.5.3 Summarizing videos using Multi-DPP

Since Multi-DPP formulation of Eq. 5.19 does not require any extra supervisory signals, it can be adopted to an optimization formula for both supervised and unsupervised training. In particular, we follow [80] in defining the similarity measure of supervised summarization approaches based on a Maximum Likelihood Estimation of the Multi-DPP measure with respect to the ground-truth labels as follows:

$$\theta^* = argmax_\theta \sum_i log\big\{P(Y^{(i)} = y^{(i)*}; L^{(i)}(\theta)\big\} \tag{5.20}$$

where θ is the set of supervised parameters, y^* is the target subset (i.e., ground-truth) and i indexes training examples.

For unsupervised summarization, we define the Multi-DPP loss based on a diversity regularization introduced in [103] that aims to only increase diversity since no summary labels are being provided.

$$\theta^* = argmax_\theta \; log\big\{P(Y; L^{(i)}(\theta)\big\} \tag{5.21}$$

where θ is the set of unsupervised parameters.

Finally we note that our supervised and unsupervised adaptations are orthogonal to other summarization approaches and can be embedded to allow any DPP-based approach (e.g., [181, 103, 14, 143, 28]) to summarize multi-stream data while preserving the temporal order and monitoring the quality of a dynamic input. Additionally, Multi-DPP is equivalent to the standard DPP decom-

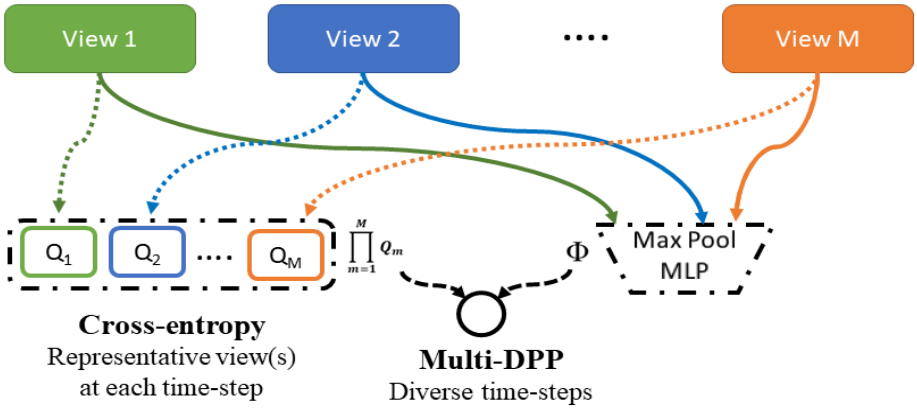

Figure 5.26: Multi-DPP is applied to increase diversity within the selected time-steps. When view labels are available, we also use cross-entropy to learn representative view(s) at each time-step.

position in Eq.1 when $M = 1$ at Eq.2. This renders Multi-DPP summarization approach as a generalization of the standard single-view summarization DPP approaches as well as orthogonal to other summarization approaches that allows them to process multi-stream data in a proper temporal order. The discussed theoretical advantage of such generalization will be further analyzed empirically at Section 5.3.

5.5.4 Summarization Framework

As shown in Figure 5.26, the input to our system is M independent views, each containing N frames. We begin by extracting spatial features of each frame in each view using a pre-trained CNN. Then, we input spatial features to a Bidirectional LSTM layer that extracts temporal features from each view. We aggregate both the spatial and temporal features, representing each frame with a comprehensive spatiotemporal feature at each view. We follow [181, 103, 14, 143] to extract spatiotemporal features using LSTM on CNN pre-computed features of the videos. We choose to share the weights of the Bi-LSTM layer across the views for two reasons: (a) it allows the system

to operate on any number of views without increasing the number of trainable parameters which alleviates overfitting, and (b) the process of learning temporal features is independent of the view, thus it should utilize data from all views to produce better temporal modeling.

We break down our objective into two tasks: selecting diverse events and identifying the view(s) contributing to illustrating each selected event in summary. In first task, to select diverse events, we construct a feature set accounting for all the views at each time-step. We do so by max-pooling the spatiotemporal features from all the views, resulting in the most prominent feature at each index of the feature vector. We follow max-pooling by a two-layer Multi-Layer Perceptron (MLP) that applies non-linear activation on joint features that are represented as Φ in Eq. 5.19.

The second task, however, is used to identify the most representative view(s) at each event. We use a two-layer MLP that classifies each view at each time step. Formulating this task as a classification problem serves three purposes. First, it selects the views that are included in the summary, which is an intrinsic part of the solution. Second, it regularizes the process of learning the importance of each event by not selecting any view when the time-step is non-important. Finally, the classification confidence of view m can be used to represent the quality (q_n^m) at time-step n. This is later used to compute the Multi-DPP measure that determines which time-steps are selected. In the case of non-overlapping views, the framework may need to select multiple views at the same time-step. That's why, we conduct an independent view classification by applying binary classification, which allows classifying each view independently from the rest.

Similar to the weights of the Bi-LSTM, the view classifier MLP weights are also shared across the views for two reasons. First, it uses the same number of trainable parameters for any number-of-views data, resulting in fewer trainable parameters which limit the problem of overfitting to training data. Second, it establishes a view-dependent classification. That is, at any time-step, choosing a representative view among all the views is affected by the relative quality of all the views, rather

than each one independently. During training, we start by estimating the quality q_n^m of each view m at each time-step n, which serves as the view selection. Then we evaluate Multi-DPP measure by merging the computed q_n^m with the joint-features Φ as in Eq. 5.19.

In our supervised setting, we optimize the view(s) selection procedure by using the binary cross-entropy objective: $-\frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N} y_n^m log(p_n^m)$; where y_n^m, p_n^m are the ground truth and model's prediction for the time-step n in view m. We jointly optimize the framework by minimizing the sum of both the losses and using the Oracle summary as the ground-truth in the supervised setting. In the unsupervised setting, view selection weights are only learned by learning the quality q_n^m from the Multi-DPP measure and we only optimize the Multi-DPP loss criterion Eq. 5.21.

Lastly, while input views are not required to be temporally aligned, they are assumed to have timestamps. This is a commonly held assumption in previous multi-view literature (e.g., [36, 74]) due to its default presence in nearly all modern recording devices. If given non-aligned views, our framework can process any number of views at each time-step since the weights of the Bi-LSTM and the MLPs are shared among the views.

5.5.5 Multi-view supervised scalability

Supervised summarization tends to have a superior generalization performance when compared to unsupervised ones, e.g., [40, 131, 181, 103]. Relying on human-annotated labels allows learning generic behavioral patterns instead of customized heuristics as in most unsupervised approaches. Nonetheless, supervision requires an abundance of labeled training data. Thus, a crucial concern of a multi-view supervised system is to be scalable in order to utilize all available forms of labels for an improved performance. Obviously, unsupervised systems do not undergo this challenge since they do not utilize labels.

In particular, a scalable multi-view video summarizer is invariant to view order and number-of-views, and therefore can learn from any data regardless of those properties. First, invariance to view order implies producing the same summary for input views (v_i, v_j, v_k) as to $(v_j, v_i, v_k); \forall i, j, k \in \{1, 2, .., M\}$, for all possible permutations of (i, j, k). Our approach satisfies this requirement by constructing joint-features via max-pooling. Thus, summary is only influenced by the most effective features with no regard to the view order.

The second condition, invariance to number-of-views, entails the ability to train on data with *varying* numbers-of-views and test on data of *any* number-of-views. Satisfying this condition requires the number of trainable parameters to be invariant from the number-of-views of the input. This way the same set of parameters can be used to train/test on data with any number-of-views. We followed two techniques ensuring a fixed number of trainable parameters: (a) max pooling view-specific features, and (b) weight-sharing for Bi-LSTM and view selection layers. Firstly, Applying max-pooling on view-specific features produces a fixed-size joint feature vector that is invariant from the number-of-views in the input. Additionally, choosing the prominent features across views entails learning intra-view dependencies. Secondly, weight sharing across Bi-LSTM view-streams and view selection layers ensures our framework has a single set of trainable parameters for each of those layers regardless number-of-views.

5.6 Experiments

5.6.1 Baseline Methods

Since our supervised approach is the first supervised multi-view summarization method, we could not compare with other supervised Multi-View approaches. Nonetheless, we compare our criterion with supervised and unsupervised single-view, and unsupervised multi-view summarizations.

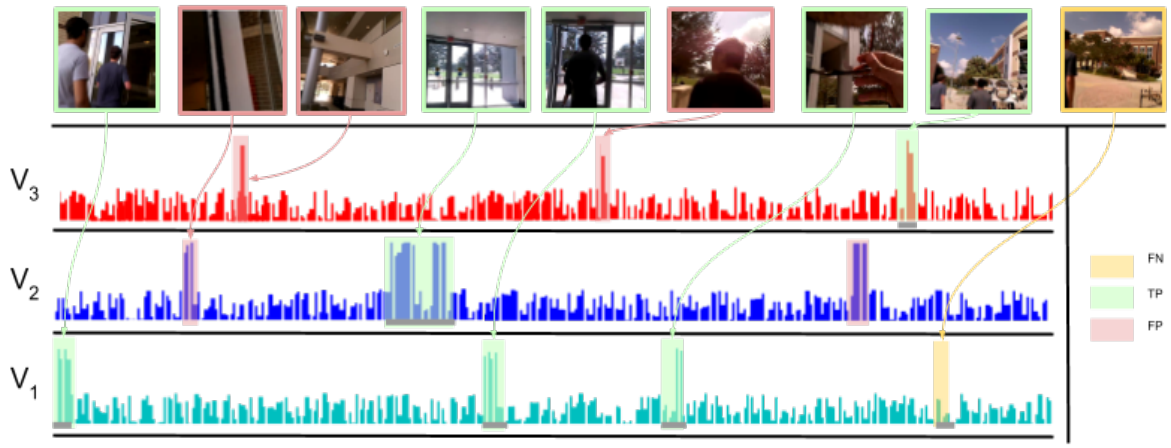

Figure 5.27: Qualitative Example of a three-view comprehensive summary, showing the confidence score of each time-step at each view. Our method may select more than one important view at the same time if they are complementary or mutually exclusive.

Additionally, we include Reinforcement Learning baselines that showed competitive performance on single-view videos.

To apply the single-view configuration on multi-view videos, we examine two settings:

- *Merge-Views*: Aggregating views then summarizing aggregate footage using a single-view summarizer. Summary is consistent if the views are independent.
- *Merge-Summaries*: Summarizing each view independently and then aggregating the summaries. Complementary to the former setting, this should result in a consistent summary if the summaries are independent.

In our experiments, we observed that the supervised version of Convolutional Sequences [139] tends to diverge when using Merge-summaries method in training due to relatively short videos in their case. Thus, we compare with the more reliable version of Merge-views. On the contrary, reinforcement learning methods [186, 187] tend to be unstable for the merge-views due to the

Table 5.13: *MultiEgo* benchmarking for two-view and three-view settings. Ours consistently outperforms the baselines on all the measures. We also run an ablation study to show the effect of optimizing the supervised Multi-DPP measure as compared to only using Cross-Entropy loss.

		Two-View			Three-View		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
Random Baseline	Uniform Sampling	9.83	10.65	9.85	5.83	5.16	5.77
Unsupervised & Sub-modular Multi-View	feature selection [116]	17.83	19.15	17.46	12.33	16.28	10.70
	joint embedding [123]	18.37	25.20	20.66	13.88	24.85	17.17
	Unpaired Data [138]	21.26	22.16	21.81	19.62	19.93	19.41
	Sub-modular [52]	19.91	25.21	22.71	18.49	22.71	20.19
Unsupervised Single-View	Adversarial [103]: Merge-Views	21.16	23.42	22.35	20.2	18.94	19.76
	Adversarial [103]: Merge-Summaries	20.61	22.05	21.12	19.32	18.24	18.96
	Convolutional [139]: Merge-Views	21.05	22.92	22.26	19.86	20.68	20.13
	Convolutional [139]: Merge-Summaries	20.64	22.34	21.87	16.52	20.47	18.91
Ours-unsupervised	**Multi-DPP**	**23.91**	**24.72**	**24.18**	**21.96**	**22.24**	**22.61**
Supervised & RL Single-View	LSTM [181]: Merge-Views	27.87	28.57	27.67	23.25	23.87	22.95
	LSTM [181]: Merge-Summaries	26.61	27.25	26.43	22.86	23.59	22.76
	Convolutional [139]: Merge-Views	26.84	26.01	26.38	22.28	23.47	22.92
	RL Diversity [186]: Merge-Summaries	25.02	27.00	25.97	23.78	22.14	23.14
	RL Classification [187]: Merge-Summaries	26.01	26.71	26.27	22.74	23.68	23.37
(Ablation Study)	Only Cross-Entropy (CE)	27.33	27.83	27.13	21.33	22.03	21.10
Ours-supervised	**Full: Multi-DPP + CE**	**28.58**	**29.05**	**28.30**	**25.06**	**25.79**	**25.03**

long sequential input where the reward is usually far away from the start of the sequence, and thus it may lead to vanishing the gradients. So, we compare with the merge-summary concatenation, where the reward function tends to be more stable. *This observed instability faced in training the baselines establishes a better motive for developing an objective like ours that is curated to be independent of number views, making it tractable during training/testing when the number of views is large, and at the same time incorporates the information from all views while preserving temporal ordering.*

5.6.2 Experimental Setup

We use GoogLeNet [157] features for all the methods as an input. For a fair comparison, we train all supervised baselines [52, 181] and Ours with the same experimental setup: iterations number,

batch size, and optimization. We note that all neural-network models have the same architecture (same number of trainable parameters) and only differ in the objective function and their training strategy to ensure a fair comparison.

The supervised frameworks are trained for twenty iterations with a batch size of 10 sequences. Adam optimizer is used to optimize the losses with a learning rate of 0.001. After each iteration, we calculate the mean validation loss and only evaluate the model with the best validation loss across all iterations. We discuss further details of the architecture and training in the appendix B.8.

As discussed in section 3.1, we categorize our dataset sequences into six collections to facilitate the training and evaluation. In our experiments, we follow a round-robin approach to train-validate-test the supervised/semi-supervised learning frameworks. We use four collections for training, one for validation, and one for testing across all the 30 different combinations of collections. Since no training is required for unsupervised approaches, we only test methods on each collection separately and report their means.

To evaluate the summaries produced by all the methods, we follow the protocols in [103, 181, 64, 148] to compare the predictions against the oracle summary. We start by temporally segmenting all views using the KTS algorithm [131] to non-overlapping intervals. Then, we repetitively extract key-shot based summaries using MAP [178] while setting the threshold of summary length to be 15% of a single view's length. For each of the selected shots, we consider all of its frames to be included in the summary.

5.6.3 Performance Evaluation

We follow [125, 123, 181, 103, 36] in using f1-score, precision, and recall to evaluate the quality of the produced summaries by comparing frame-level correspondences between the predicted

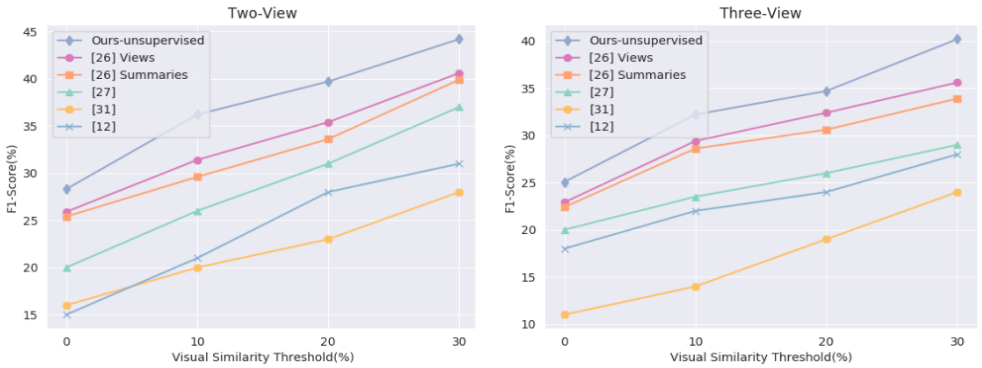

Figure 5.28: F1-score computed whereas unsupervised prediction models are not penalized if mistakenly chose a view that is similar to GT view within various threshold levels.

summary and the ground-truth summary. Table 5.13 shows the mean precision, recall, and F1-score across all the combinations of training-validation-testing for both the two-view setting and three-view setting (i.e., stages two, three of the annotations).

In general, supervised frameworks perform better than unsupervised ones due to learning from human annotations. For unsupervised methods, [123, 116, 52, 138] obtain the lowest performance indicating their inability to adapt to visual changes occurring in egocentric motion due to the lack of summary labels. However, using adversarial training [103] seems to improve the results even with a single-view setting since the learning distribution converges to true data distribution, and it better learns to isolate egocentric-noise. Similarly, the supervised single-view BiLSTM [181] and Convolutional Sequences [139] reasonably adapt to egocentric visual noise utilizing the summary labels. Only our model monitors the egocentric-introduced noise and process data in a proper temporal order, achieving the best performance in both unsupervised and supervised comparisons.

To study the impact of enforcing diversity, we run an ablation study by evaluating our supervised approach with only optimizing cross-entropy loss(Ours: Cross-Entropy (CE) in Table 5.13). This

Table 5.14: F1-Score of fixed-cameras multi-view benchmarking. We train our supervised model on **Multi-Ego** and test it on three datasets.

Method	Office	Campus	Lobby
Graph [127]	41.3	49.1	73.4
RandomWalk [36]	75.8	61.6	86.8
RoughSets [86]	75.8	62.1	84.2
BipartiteOPF [74]	81.8	71.8	88.2
Unpaired Data [138]	91.0	80.5	89.3
Joint embedding [123]	89.4	77.8	92.5
Convolutional [139]-Unsup	90.2	78.6	92.5
Convolutional [139]-Sup	94.0	81.9	93.0
RL Diversity [186]	92.9	80.6	91.4
RL Classification [187]	92.1	82.5	92.2
Ours-unsupervised	90.7	81.2	92.7
Ours-supervised	**94.2**	**86.1**	**93.4**

corresponds to training our model by only selecting representative views, without explicitly enforcing diversity. Evidently, adding Multi-DPP measure to the CE loss improves the results, especially in the three-view setting due to the increase of input footage required to diversify. It is worth noting that using only Multi-DPP is equivalent to our unsupervised version.

Generally, it can be noticed that performance in the two-view setting is higher than that in the three-view setting, although methods' ranking remains the same. This is because of the increase in problem complexity when considering more views to be summarized, causing the performance to drop. Additionally, the performance gap increases

as we move from two-view to three-view setting. Theoretically, we expect approaches such as [139, 181, 103, 186] drop performance as the number of views grows and this is backed up empirically. Secondly, whether we concatenate views or concatenate summaries in order to adapt [181, 186, 139, 103], the complexity of the adaptation is unnecessarily high (either a larger DPP kernel in case of view concatenation and processing each view separately in summary concatena-

Table 5.15: Scalability Analysis: Our framework can be trained and tested on data of different number-of-views. It utilizes data from various number-of-views to improve the performance on test data.

Test	Train	Precision	Recall	F1-Score
two-view	2×two-view	29.83	29.77	29.67
	3×three-view	29.77	30.30	30.2
	2×two-view + 3×three-view	**34.37**	**35.03**	**34.33**
three-view	2×three-view	18.53	18.80	18.33
	2×two-view	18.23	18.27	17.67
	3×two-view + 2×three-view	**21.53**	**21.87**	**21.33**

tion scenario). Our proposed approach uses a maxpool operation as well as view quality multiplication to effectively represent all views while preserving the computational/memory efficiency.

Additionally, we address a shortcoming of the common evaluation metrics that present itself in our setting. Consider the case of two or more views having nearly identical visual content at the same time-step, which happens due to the dynamic overlap of fields-of-view. When annotating the sequences, the user will only include one of the views in the ground-truth summary at important events. However, if the prediction model selects any of the other views, it should not be penalized since the views are visually similar. To address this case, we evaluate the F1-score at several levels of similarity thresholds. That is, if the Euclidean distance of the normalized CNN features between two views at the same time-step is less than a threshold (0%, 10%, 20%, 30%), we do not penalize the prediction model if it selects any of the views instead of the other. We recompute the F1-scores for all unsupervised models at different threshold values. As shown in Fig. 5.28, our method continues to obtain the highest F1 at all threshold levels.

Finally, we investigate the performance of our approach on fixed-cameras multi-view setting,

which is a special case of our generic configuration. We evaluate our model on three standard fixed-cameras multi-view benchmarks: Office, Campus, and Lobby datasets [36, 120]. We train our supervised model on our *Multi-Ego* dataset, and evaluate it on the testing dataset. Table 5.14 shows a substantial success in transferring the learning from one domain (egocentric multi-view) to another domain (static multi-view) without the need to specifically-tailored training data. Thus, we provide the first supervised multi-view summarization that significantly outperforms state-of-the-art unsupervised approaches while only being trained on our data. Additionally, our unsupervised model outperforms them due to explicitly enforcing diversity and quality constraint. The consistent advantage in the three experimental environments for both our supervised and unsupervised models demonstrates the versatility of the proposed approach in handling static/egocentric videos in a generic summarization setting.

5.6.4 Supervised Scalability Analysis

In this section, we study our supervised framework's capability to learn from a varying number-of-views in a sequence by verifying if the training process can exploit any increase in data regardless of its numbers-of-views. We start by splitting our data into two categories of nearly the same number of sequences: (a) three-view (Collections: Indoors-Outdoors, SeaWorld, Supermarket), and (b) two-view (Collections: Car-Ride, College-Tour, Library). We investigate the performance of three train/test configurations where testing data is limited to a single category:

1. Same category training (2×two-view& 1×two-view): Train on 2 collections from same category as testing.

2. Different category training (3×two-view& 3×three-view): Train on 3 collections from one category, and then test it on a collection belonging to a different category.

3. Training using Data from the two categories (3×two-view + 2×two-view& 2×two-view +

3×two-view): Train on data from different categories, and test it on a collection from one of the categories in the training data.

For each of the scenario enumerated above, the model is tested on all the three possible test collections available to us. For example, when evaluating 3×two-view, there are three collection instances of the three-view category. Therefore, we report average performance across all them.

As shown in Table 5.15, training our framework on same categories or different categories obtain comparable results when testing on both two-view and three-view settings. However, increasing training data size by combining both categories significantly improves the results. This shows that our model can be trained and tested on data of various number-of-views and also is able take advantage of any data increase with no regard to its number-of-views setting.

5.7 Conclusion

In this chapter, we proposed the problem of multi-view video summarization for dynamically moving cameras that often do not share the same field-of-view. Our formulation provides the first supervised solution to multi-stream summarization in addition to an unsupervised adaptation. Unlike previous work in multi-view video summarization, we presented a generic approach that can be trained in a supervised or unsupervised setting to generate a comprehensive summary for all views with no prior assumptions on camera placement nor labels. It identifies important events across all views and selects the view(s) best illustrating each event. We also introduced a new dataset, recorded in uncontrolled environments including a variety of real-life activities. When evaluating our approach on the collected benchmark and additional three standard mutli-view benchmark datasets, our framework outperformed all baselines of state-of-the-art supervised, reinforcement and unsupervised single- and multi-view summarization methods.

CHAPTER SIX: CLOSING REMARKS

The complex nature of real world data; the multi-modal inherent structure and the predominance of cluttered insignificant patterns, makes modeling true-data distributions quite an intricate problem. Subsequently, training a neural network to learn from such compound distributions must be governed by a learning prior that emphasizes critical patterns and disregards inconsequential information. In this work we discussed few discrepancies encountered when attempting to use neural networks in modeling physical world's data. Namely, we examine the prevalent multi-modal nature of true-data distribution and propose a method to learn the inherent diversity structure of training data as a prior. Additionally, we examined utilizing an auxiliary data domain as well as an auxiliary learning task to improve the performance of the primary objective when learning on data sampled from the original domain. Finally, we investigated the shortcoming of using a limited-scope objective in training neural networks and demonstrated an instance of generalizing that objective to adopt the generic case of training. In our experiments, we considered a wide variety of applications to establish those discrepancies, ranging from image retrieval and generation, to video summarization and actionness ranking. For each discrepancy, we established drawbacks of trivially training a network and the advantage brought by introducing the corresponding learning prior.

Despite the merit of conditioning a learning prior in mitigating those discrepancies, some concerns are realized that need further study. A multitude of issues can prevent proper learning due to the apparent disparity between physical world's representation and the empirical samples provided for training the neural network. For example, the efforts needed to close a wide domain gap to fully utilize cross-domain training samples. Another concern is properly adjusting multi-task learning frameworks to employ the task generalization capability and use supporting tasks to enhance the primary objective. Thus, in our future work, we plan to shed more light on those concerns and examine further discrepancies within the training and how to mitigate them.

APPENDIX A: DIVERSE SAMPLING

7.1 Experimental Details

7.1.1 Architecture

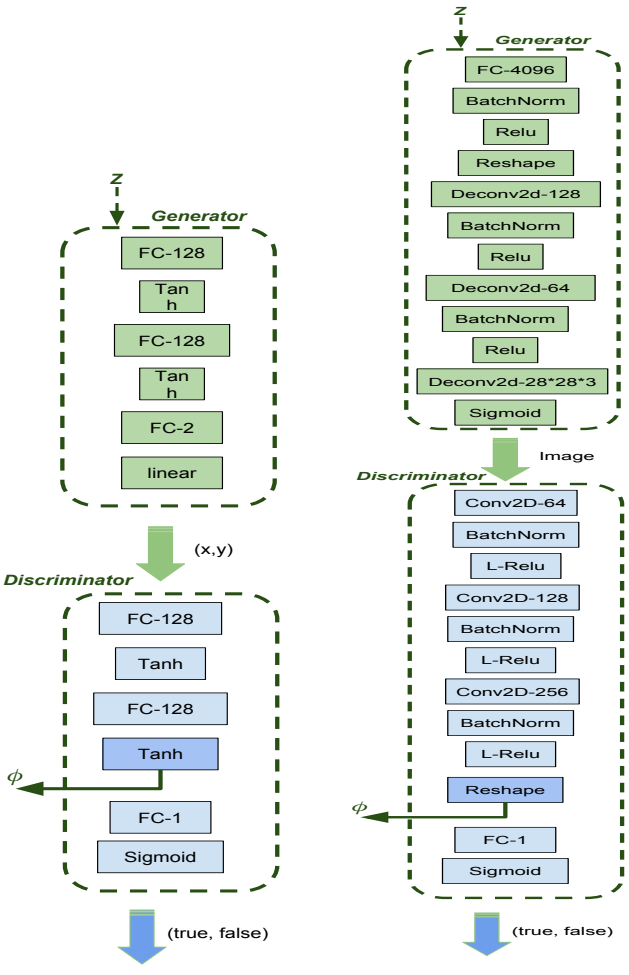

Figure 7.29: Architecture employed in (a) Synthetic experiments. (b) Stacked-MNIST and CIFAR-10 experiments.

7.1.2 Hyperparameters

In all of our experiments, we use Adam Optimizer with $\beta_1 = 0.5$ and $\epsilon = 1 \times 10^{-8}$. For the synthetic data experiments, we follow the configurations used by [151] and [110]. We use 1×10^{-4} for the discriminator learning rate, and 1×10^{-3} for the generator learning rate. For synthetic data, we use a batch size of 512. For Stacked-MNIST and CIFAR-10 we use a batch size of 64. For CelebA, we use a batch size of 16.

For the Stacked MNIST, CIFAR-10 and CelebA datasets, we use 2×10^{-4} as the learning rate for both of the generator and the discriminator. To relatively stabilize the training of DCGAN, we follow the protocol in [45] to train it by applying a learning rate scheduler. The decay is to happen with a ratio of $1/(\#max - iters)$ at every iteration.

In our experiments, we used the official implementations of: WGAN [8][3], WGAN-GP [45][4], DCGAN [132][5], ALI [24][6], VEEGAN [151][7] and DeLiGAN [48][8].

7.2 Synthetic Data Collections

The first data collection is introduced in [110] as a mixture of eight 2D Gaussian distributions arranged in a ring. This distribution is the easiest to mimic since it only requires the generated data to have an equal repulsion from the center of the distribution, even if it is not targeted to the modes. The second and third collections were introduced by [151]. In the second collection,

[3] https://github.com/martinarjovsky/WassersteinGAN

[4] https://github.com/igul222/improved_wgan_training

[5] https://github.com/carpedm20/DCGAN-tensorflow

[6] https://github.com/IshmaelBelghazi/ALI

[7] https://github.com/akashgit/VEEGAN

[8] https://github.com/val-iisc/deligan

there is a mixture of twenty-five 2D Gaussian distributions arranged in a grid. Unlike the first collection, this one requires a more structured knowledge of the true data modes' locations. The last collection is a mixture of ten 700 dimensional Gaussian distributions embedded in a 1200 dimensional space. This mixture arrangement mimics the higher dimensional manifolds of natural images and demonstrates the effectiveness of each method on manipulating sparse patterns.

7.3 Additional Experiments

7.3.1 Invariance to Poor Initialization

Since the weights of the generator are being initialized using a random number generator $\mathcal{N}(0, 1)$, the result of a generative model may be affected by poor initializations. In Figure 7.30 we show qualitative examples on 2D Grid data, where we use high standard deviation for the random number generator $(i.e., \sigma > 100)$ as an example of poor initializations. Evidently, GDPP-GAN attains the true-data structure manifold even with poor initializations. On the other extreme, WGAN-GP tends to map the input noise to a disperse distribution covering all modes but with low-quality generations.

7.3.2 [151] Experimental Setting on Real Data

To show the robustness of our approach to the experimental setting, we further examine it under another more challenging setting. The setting described in [151] entails an architecture and hyperparameters that produce relatively poor results as compared with the setting of Table 3. For example, In [151] setting, DCGAN produces 99 modes, while in our experimental setting, DCGAN produces 427 modes on Stacked MNIST dataset. We note that our main results in Table 3 are computed using the same experimental setting suggested by [45] and [110] on a more realis-

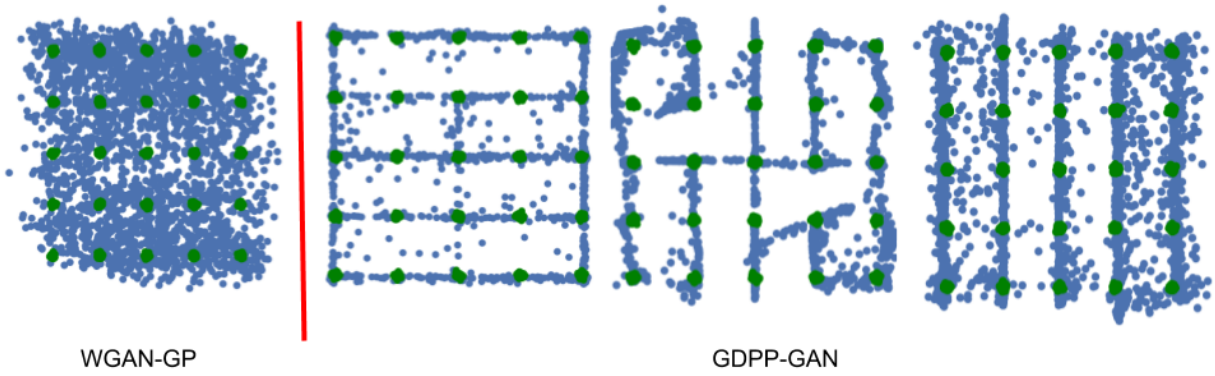

Figure 7.30: The effect of poor initialization on generations: GDPP-GAN models true manifold structure even with poor initializations, while WGAN-GP maps noise to disperse distribution covering the modes with low quality samples.

tic architecture. Our method remains to have a clear advantage when compared to the rest of the baselines for both CIFAR-10 and Stacked-MNIST (e.g., covering 90.6% more modes on Stacked-MNIST from 150 to 286 and at a higher quality). We obtain the first four rows from [151].

7.3.3 Eigendecomposition Running time

Eigendecomposition of an $n \times n$ matrix requires $O(n^3 + n^2 \log^2 n \log b)$ runtime within a relative error bound of 2^{-b} as shown in [122]. In our GDPP loss, we perform two eigendecompositions every training iteration: L_{S_B}, L_{D_B} corresponding to the fake and true DPP kernels respectively. Therefore, the run-time analysis of our loss is $O(n^3)$ at every iteration, where n is the batch size.

Normally the batch size does not exceed 1024 for most training paradigms due to memory constraints. In our experiments, we used 512 for synthetic data and 64 or 16 for real data. Hence, the eigendecomposition does not account for a significant delay in the method.

To further verify this claim, we measured the relative time that eigendecompositions take of each

Table 7.16: Performance on real datasets using the challenging experimental setting of [151]. GDPP-GAN continues to outperform all baselines on both Stacked-MNIST and CIFAR-10 for all metrics.

	Stacked-MNIST		CIFAR-10
	#Modes (Max 1000)	KL div.	IvO
DCGAN [132]	99	3.4	0.00844
ALI [24]	16	5.4	0.0067
Unrolled-GAN [110]	48.7	4.32	0.013
VEEGAN [151]	150	2.95	0.0068
GDPP-GAN (Ours)	**286**	**2.12**	**0.0051**

iteration time. We obtained 11.61% for Synthetic data, 9.36% for Stacked-MNIST data and 8.27% for CIFAR-10. Additionally, Table 4 in the original text shows that our method obtains the closest running time to the standard DCGAN, and is faster than the rest of baselines by a large margin (e.g., 5.8$\times$ faster than WGAN-GP). The large margin in running time between GDPP and WGAN-GP [45] is attributed to two factors. First, WGAN-GP trains the discriminator several times for every one iteration of the generator, which significantly increase the running time. Second, WGAN-GP calculates the gradients of the discriminator at every iteration, which is a very computationally expensive operation. On the other hand, GDPP does not alter the adversarial learning paradigm and only adds an insignificant computation overhead as discussed.

7.3.4 Number of statistically-Different bins (NDB)

[136] proposed to use a new evaluation metric to assess the severity mode collapse severity in a generative model. They based their metric on a simple observation: In two sets of samples that represent the same distribution, the number of samples that fall into a given bin should be the same up to a sampling noise. In other words, if we clustered the true-data distribution and fake-data distribution to the same number of clusters/bins, then the number of samples from each

distribution in every bin should be similar.

We follow [136] to compute this metric on MNIST [81] dataset, and compare our method with their results in Table 7.17. We note that we used their open-source implementation of the metric, and we obtained the first three rows from their paper. We use 20,000 samples from our model and the MNIST training data to compute the NDB/K.

Table 7.17: NDB/K - numbers of statistically different bins, with significance level of 0.05, divided by the number of bins K.

Model	K =100	K =200	K =300
Train	0.06	0.04	0.05
MFA [136]	0.14	**0.13**	0.14
DCGAN [132]	0.41	0.38	0.46
WGAN [8]	0.16	0.20	0.21
GDPP-GAN	**0.11**	0.15	**0.12**

7.4 Additional Qualitative Results

Random samples generated on Stacked-MNIST. GDPP-GAN converges faster than GDPP-VAE and generates sharper samples.

Random samples generated by GDPP-GAN and WGAN-GP respectively in an unsupervised setting. The generations are qualitatively similar while GDPP-GAN outperforms WGAN-GP quantitatively even though it was trained for half the number of iterations.

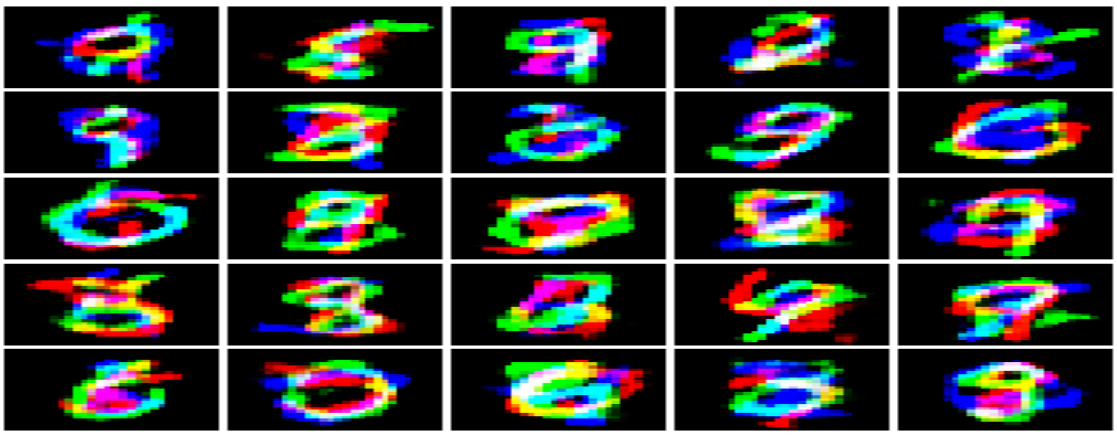

Figure 7.31: GDPP-GAN after 15K iterations.

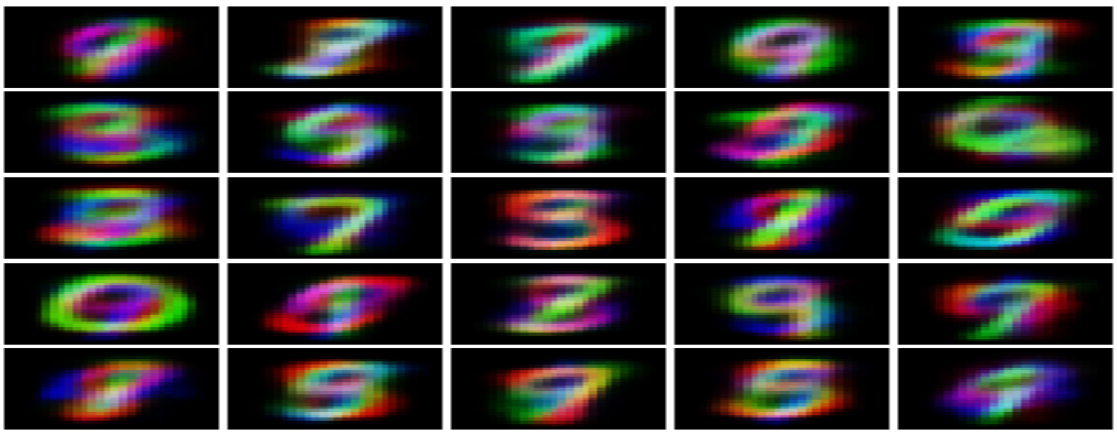

Figure 7.32: Generations by GDPP-GAN after **100K** iterations.

Figure 7.33: Generations by GDPP-GAN after **100K** iterations.

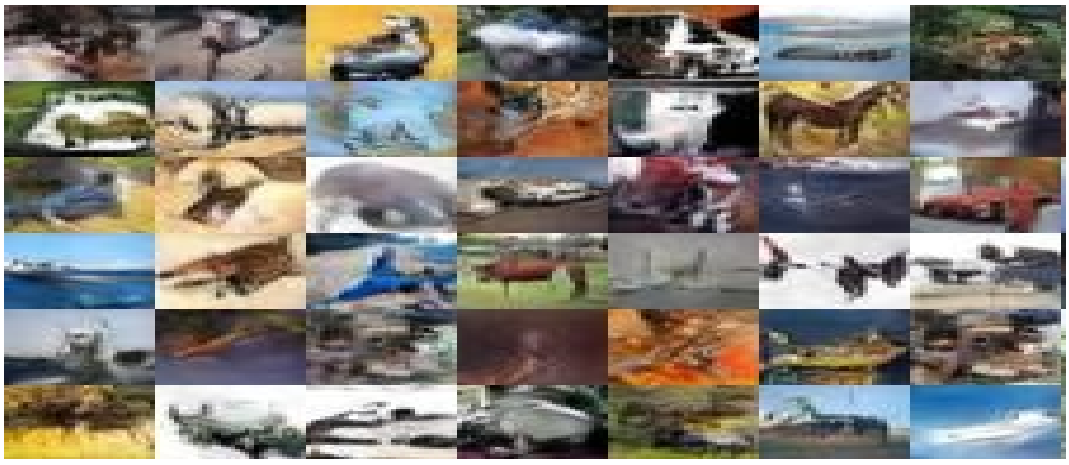

Figure 7.34: Generations by WGAN-GP after **200K** iterations.

Figure 7.35: Fixed noise qualitative progression for different models. GDPP-GAN starts synthesizing realistic generations the first with more diverse patterns than both DCGAN and WGAN-GP.

(a) random samples using WGAN-GP

(b) random samples when adding GDPP loss

Figure 7.36: Comparing [68] without (a) and with our loss (b) after 200,000 training iterations.

APPENDIX B: OBJECTIVE GENERALIZATION

As been explained in the original text, our major contributions can be summarized in four points:

1. Addressing the generalized problem of multi-stream dynamic video summarization where the input consists of multi-view sequences recorded by dynamically moving cameras that intermittently share their field-of-views. The output is a holistic summary that represents important and representative events across all of the views.

2. Since we are the first to address this problem, no testing data is applicable. Thus, we introducing a new dataset that includes a variety of real-life activities in uncontrolled environments, while altering people, actions, and places.

3. Collecting user annotations for the dataset, and running a comprehensive analysis to ensure reliability and consistency.

4. Proposing a new generic approach that operates on multi-view videos by identifying the important events across all the views as well as selecting representative view(s) that can contribute to illustration of the events in the summary. Our method can be used in a supervised or unsupervised setting and applicable to fixed-cameras or dynamic-cameras footage.

This appendix is organized as follows. In section 2, we present more details about the dataset presented in the main text. In section 3, we elaborate on the annotation procedure and illustrate some qualitative results. In section 4, we conduct further analysis on the collected annotation to help us understand the behavior of human annotators, that ultimately should help us for a better understanding of the problem. Finally, in section 5 we provide additional implementation details of our framework to help reproducing our results.

We note that we will be sharing our dataset and materials to facilitate future research in this area.

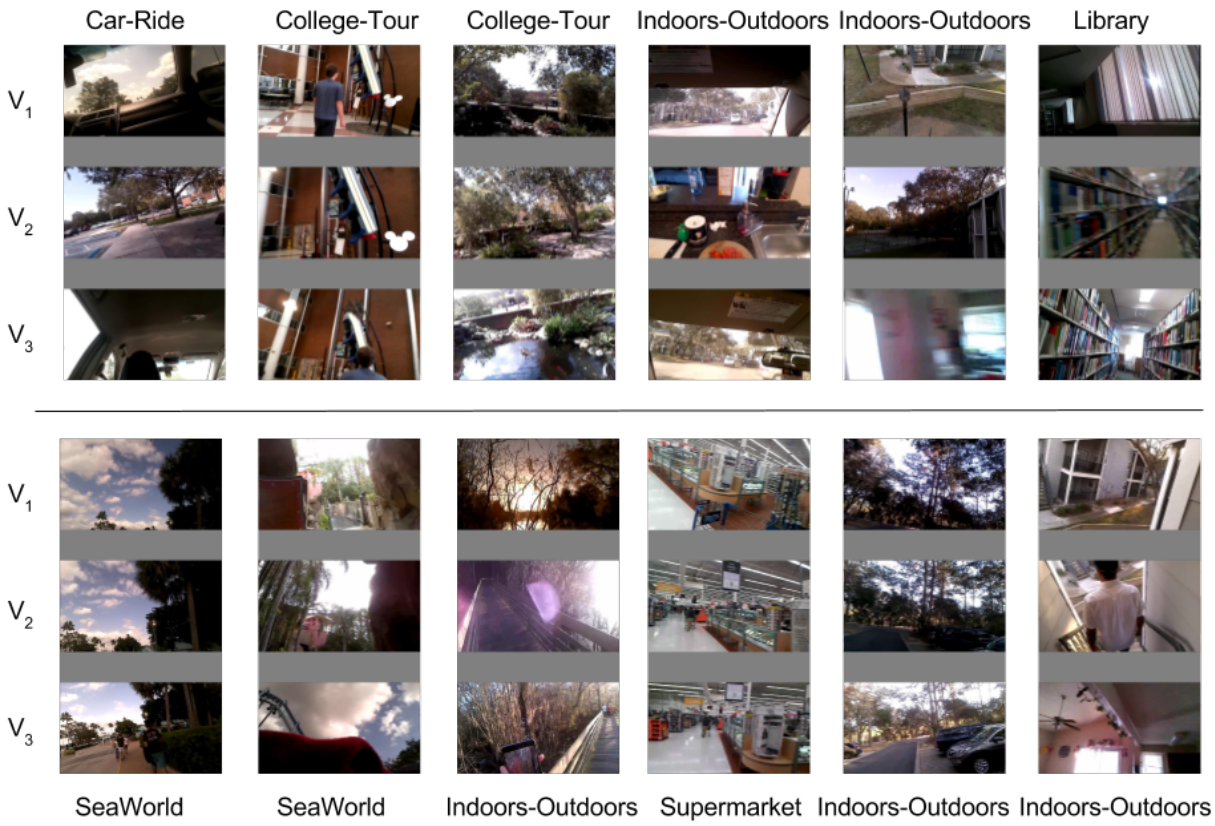

Figure 8.37: Sample frames from the dataset.

8.1 Dataset Description

We collected a total of 12 hours of videos per view for three people performing different activities. During collecting the videos we varied the environments, people, and activities to ensure a reasonable diversity that occurs in a day-to-day activities. Figure 8.37 shows sample frames from the sequences covering all the collections. As shown by the samples, the views are dynamically changing their perspective and often don not share the same field-of-view. Also, using egocentric cameras causes the videos to contain rapid changes in illumination, unpredictable camera motion, unusual composition, and often complex hand-object manipulations.

From all the videos, we extracted 41 sequences that contain the highest diversity in terms of actions and environments. Each sequence contains three views that are temporally aligned across the views (i.e., all the views have the same starting and ending points). The sequences varied in length from 3 to 7 minutes and have been recorded using cameras that have frame rate of 30fps. To facilitate training and evaluating the sequences, we grouped the sequences into 6 collections that represent different scenarios:

- **Car-Ride:** 1 driver and 3 passengers are involved, the driver and two passengers hold the cameras. They all ride a car. The driver drops the passengers off separately in different locations, then comes back and picks them up again. This process is repeated several times in various locations and different orders. At the end, they park the car in a garage and enter a building. The videos were recorded in daytime.
- **Library:** 3 students go to a library. They walk around the library, search for books for the most part. Occasionally, they stop to take a glance at an interesting book, or for a brief talk when they run into each other.
- **Supermarket:** 3 people walk around a supermarket for grocery shopping. Sometimes they stop, examine or pick interesting products and goods. Occasionally they stop and talk to other people.
- **College-Tour:** 1 tour guide and 3 visitors are involved. The guide and two visitors hold the cameras. The guide walks them through the campus explaining most of the locations in daytime. Occasionally they stop and the guide gives them brief details about an attraction. They walk through the buildings in the campus and outside attractions.
- **Indoors-Outdoors:** Several family members, three of them hold cameras. They perform different activities such as cooking, playing cards, walking outside the house, learning driving the car, going to a park, playing Xbox games, etc.

Table 8.18: Statistics of the Dataset

Collection Name	Number of sequences	Number of frames per sequence
Car-Ride	4	360, 360, 360, 360
College-Tour	11	190, 300, 270, 300, 240, 280, 260, 210, 300, 240, 240
Library	5	225, 300, 300, 300
Supermarket	9	270, 270, 230, 225, 210, 315, 300, 270, 210
Indoors-Outdoors	8	300, 240, 300, 300, 345, 300, 225, 270
SeaWorld	4	300, 300, 260, 300
Total	**41**	**11,135 x 3 views = 33,405**

- **SeaWorld:** 3 friends go to a sea-world show in daytime. The sequences includes activities such as driving, walking in the sea-world, checking in through the gate, watching the show, taking photos of each others, etc.

Table 8.18 shows the number of sequences per collection and the number of frames per sequence; after down-sampling the frame rate to 1 fps.

8.2 Annotation Procedure

As mentioned in section 2, the original frame rate of the cameras used in recording is 30 fps. To generate a human-accessible data, we uniformly subsampled the frames to 1 fps. For each view, we generated non-overlapping shots from every consecutive 3 frames in the data. We chose to include 3 frames per shot empirically such that each shot contains a consistent action per view. Figure 8.38 shows sample shots (3-consecutive frames) from the sequences for all the views.

We asked five human users (4 undergraduate students, and 1 high-school student) to create annota-

Figure 8.38: Sample Shots (3-Consecutive Frames) from the datasets.

tions for all the sequences. Even though, subsampling the frames resulted in 180 to 360 frames per view in each sequence, but it is still a considerable number to show to human annotators. This may cause subjects to forget the details within a view or across the views. To remedy this, we displayed the shots to the users and asked them to select the minimal number of shots that best represent the videos as a summary.

Since the story unfolding dependencies are complex within the views, as well as across the views, we decided to use *annotating in stages* procedure. In *stage one*, the users were asked to track the story unfolding within each view and summarize them independently regardless of the correlation

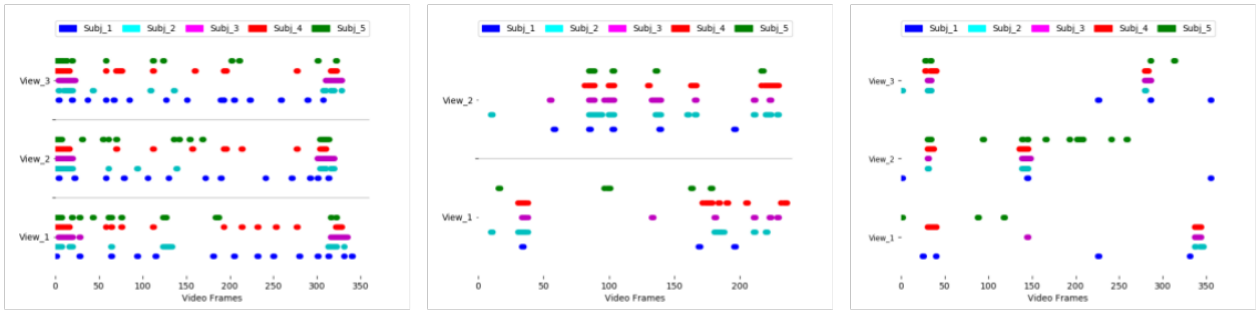

Figure 8.39: Visualizations sample of users summaries in the three stages.

among the views. In *stage two*, the users were asked to track the story unfolding within a pair of views and select the most important events across all the views, and in each important event to pick the most representative view for such event. Similarly in *stage three*, the users were asked to summarize all the views simultaneously such that all the important events from all the views as well as the most representative view(s) for each event are considered.

Using the ”annotating in stages” technique helps the users understand the dependencies within each view independently in stage 1, then in stages 2 & 3 they start to develop better understanding and capturing the dependencies across the views. It is worth noting that the annotators were not limited to choose only one representative view for stages 2& 3 of a certain shot. They could choose as many views they deem representative for an event as long it constitutes a minimal length summary.

After collecting the annotations, we ran a consensus analysis on the annotations to ensure a reliable and consistent set of annotations for all the stages. As reported in the main text, we computed the average pairwise f1-measures as well as the selection ratio metrics. We find that there is a substantial consensus between the users. We also plotted a visualizations for frames selected by all the annotators to further show a qualitative verification of the consensus. Figure 8.39 shows sample visualization of the annotations in the three stages.

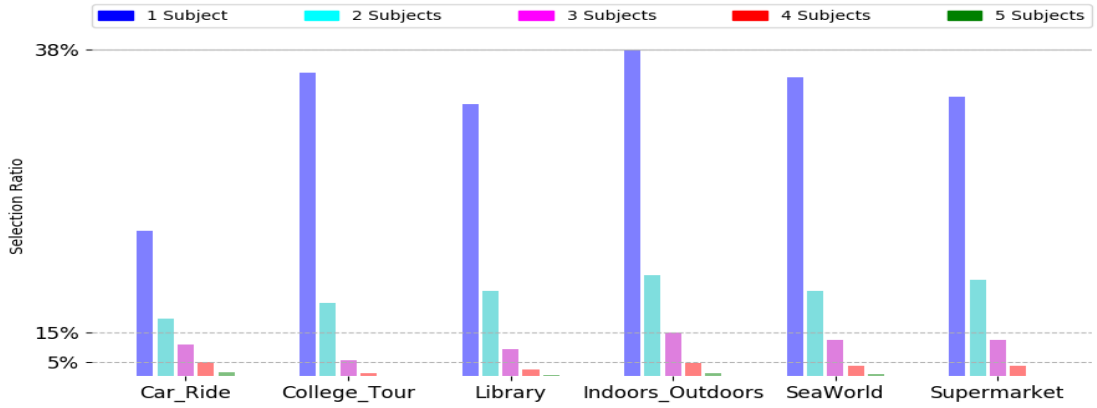

Figure 8.40: Percentage of frames selected by at least 1, 2, 3, 4, 5 subjects for the annotations. In every collection, at least 3 annotators agree on $5-15\%$ which represents the summary.

For further annotation quality assessment, we used the selection ratio metric. According to [148, 40, 142], the usual summary length should be 5-15% of the total length of the sequence. Any frame that is a part of the final summary should be selected by at least three out of the five annotators. Figure 8.40 shows the ratio of the frames (with respect to sequence length) that have been chosen by at least 1, 2, 3, 4, and 5 subjects, respectively for each collection in stage three. For all the collections, the ratio of the frames chosen by at least three users is within the 5-15% range.

8.3 Additional Analysis

Can we explicitly see instances where annotators choose frames from one view over the other consistently? To answer this question we ran the following experiment. First, we identified the *conflict shots* which are the shots selected from different views at the same time step in 1-view summaries (i.e., present similar information). Then we calculated the frequency of each view for those time-steps in the 2-view oracle summary which are shown for the oracle summary(i.e., GT) and our method's summary in figures 8.41 and 8.42 respectively. Evidently our method's summary

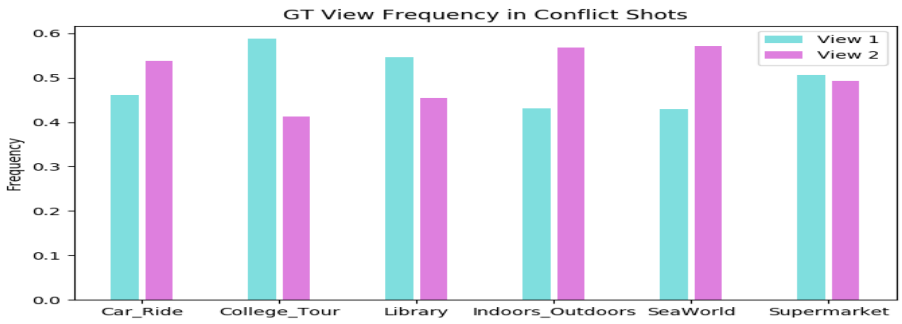

Figure 8.41: Conflict shots frequency in GT.

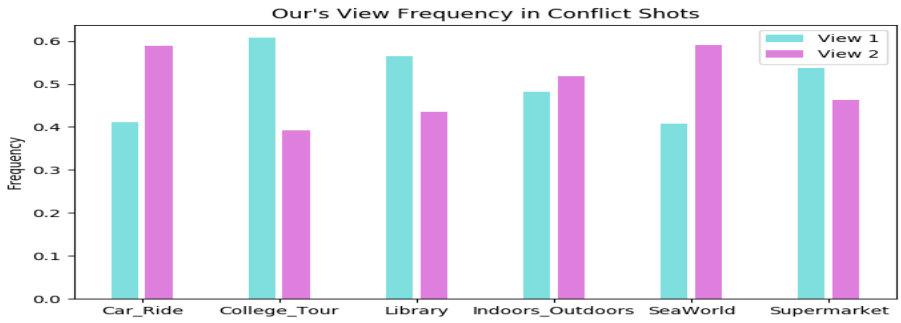

Figure 8.42: Conflict shots frequency in Ours

resembles the distribution of the ground-truth.

8.4 Implementation

We used Tensor-Flow framework for our system implementation. Using the oracle summary as a ground-truth (Section 3.3 in the main text), we construct two types of labels to match our two network outputs. Those labels are used to train the network in the supervised setting. The first type is the view-selection importance used to train the quality score q_n^m, discussed in Section 4.3 of the original text. The second type of labels is the time-step importance. Both types are used to compute

the MLE of Multi-DPP criterion and the normalized cross-entropy loss of the view-selection MLP which is equivalent to the oracle summary labels.

As explained in section 4.4 of the main text, the network is designed to have a number of trainable parameters that is invariant to the number of views in the data. Therefore, we are sharing the LSTM weights among all the units across all the views. We also share the weights of the view-selection MLPs across all the views. Max-pooling is applied to extend the joint features from all the views. As illustrated in section 4.3 of original text, We apply two MLPs to utilize DPP quality-diversity decomposition: Q_v and Φ. We use 256 hidden units for all the LSTM and MLP units. The view-classifier as well as the DPP diversity decomposition MLP contain two hidden layers.

We use a tanh activation layer for the LSTM units and as hidden activations for the MLPs. Additionally, we use a sigmoid activation for the view-selection classifier and a linear activation for the time-step feature MLP Φ. View-selector MLP outputs a scalar value for each view at each time-step. However, Φ MLP outputs a joint feature vector of size 256 at each time-step.

For evaluation, we used the code provided by [148, 49] that is also used by [181, 103, 64]. We modified the code to match our multi-view dataset and sat the threshold of the summary length to be 15% of the single-view length.

LIST OF REFERENCES

[1] S. Ardeshir and A. Borji. Ego2top: Matching viewers in egocentric and top-view videos. In *European Conference on Computer Vision*, pages 253–268. Springer, 2016.

[2] S. Ardeshir and A. Borji. An exocentric look at egocentric actions and vice versa. *Computer Vision and Image Understanding*, 2018.

[3] S. Ardeshir and A. Borji. Integrating egocentric videos in top-view surveillance videos: Joint identification and temporal alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 285–300, 2018.

[4] S. Ardeshir and A. Borji. Egocentric meets top-view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[5] S. Ardeshir, K. Malcolm Collins-Sibley, and M. Shah. Geo-semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2792–2799, 2015.

[6] I. Arev, H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)*, 33(4):81, 2014.

[7] B. Ariel, W. A. Farrar, and A. Sutherland. The effect of police body-worn cameras on use of force and citizens' complaints against the police: A randomized controlled trial. *Journal of quantitative criminology*, 31(3):509–535, 2015.

[8] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *stat*, 1050:26, 2017.

[9] M. P. R. C. R. M. Betancourt A. The evolution of first person vision methods: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 2015.

[10] A. Borodin and A. Soshnikov. Janossy densities. i. determinantal ensembles. *Journal of statistical physics*, 113(3-4):595–610, 2003.

[11] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles. Sst: Single-stream temporal action proposals. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6373–6382. IEEE, 2017.

[12] Z. Cernekova, I. Pitas, and C. Nikou. Information theory-based shot cut/fade detection and video summarization. *IEEE Transactions on circuits and systems for video technology*, 16 (1):82–91, 2006.

[13] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *ICLR*, 2017.

[14] B.-C. Chen, Y.-Y. Chen, and F. Chen. Video to text summary: Joint video summarization and captioning with recurrent neural networks. In *BMVC*, 2017.

[15] B.-C. Chen, Y.-Y. Chen, and F. Chen. Video to text summary: Joint video summarization and captioning with recurrent neural networks. 2017.

[16] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 748–755, 2014.

[17] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[19] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25, 2016.

[20] D. Davidson. Actions, reasons, and causes. *The journal of philosophy*, 60(23):685–700, 1963.

[21] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, and A. de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[23] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *ICLR*, 2017.

[24] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *ICLR*, 2017.

[25] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *ICLR*, 2017.

[26] M. Edraki and G.-J. Qi. Generalized loss-sensitive adversarial learning with manifold margins. In *ECCV*, 2018.

[27] M. Elfeki and A. Borji. Video summarization via actionness ranking. *Winter Applications in Computer Vision (WACV)*, 2019.

[28] M. Elfeki, C. Couprie, M. Riviere, and M. Elhoseiny. Gdpp: Learning diverse generations using determinantal point process. *arXiv preprint arXiv:1812.00068*, 2018.

[29] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[30] C. Fan, J. Lee, M. Xu, K. K. Singh, Y. J. Lee, D. J. Crandall, and M. S. Ryoo. Identifying first-person camera wearers in third-person videos. *arXiv preprint arXiv:1704.06340*, 2017.

[31] C. Fan, J. Lee, M. Xu, K. K. Singh, Y. J. Lee, D. J. Crandall, and M. S. Ryoo. Identifying first-person camera wearers in third-person videos. *arXiv preprint arXiv:1704.06340*, 2017.

[32] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011.

[33] R. J. Fathi A, Farhadi A. Understanding egocentric activities. *Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE*, 2011.

[34] R. J. Fathi A, Li Y. Learning to recognize daily actions using gaze. *Computer Vision–ECCV*, 2012.

[35] Y. Feng, Y. Yuan, and X. Lu. Learning deep event models for crowd anomaly detection. *Neurocomputing*, 219:548–556, 2017.

[36] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou. Multi-view video summarization. *IEEE Transactions on Multimedia*, 12(7):717–729, 2010.

[37] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals, 2017.

[38] G. Geisler, G. Marchionini, B. M. Wildemuth, A. Hughes, M. Yang, T. Wilkens, and R. Spinks. Video browsing interfaces for the open video project. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 514–515. ACM, 2002.

[39] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. Torr, and P. K. Dokania. Multi-agent diverse generative adversarial networks. *CVPR*, 2018.

[40] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2069–2077. Curran Associates, Inc., 2014.

[41] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pages 2069–2077, 2014.

[42] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2014.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[44] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[45] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of Wasserstein GANs. In *NIPS*. 2017.

[46] S. Gupta. 1 determinantal point processes.

[47] S. Gupta. Determinantal point processes. *Lecture notes*, 2015.

[48] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu. Deligan: Generative adversarial networks for diverse and limited data. In *CVPR*, pages 4941–4949, 2017.

[49] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[50] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[51] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings CVPR 2015*, pages 3090–3098, 2015.

[52] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3090–3098, 2015.

[53] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527. URL `http://dx.doi.org/10.1162/neco.2006.18.7.1527`.

[54] R. Hirsch. *Seizing the Light: A Social & Aesthetic History of Photography*. Taylor & Francis, 2017.

[55] R. Hirsch. *Seizing the Light: A Social & Aesthetic History of Photography*. Taylor & Francis, 2017.

[56] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[57] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[58] K. Hong and A. Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, 2014.

[59] R. Hong, L. Li, J. Cai, D. Tao, M. Wang, and Q. Tian. Coherent semantic-visual indexing for large-scale image retrieval in the cloud. *IEEE Transactions on Image Processing*, 2017.

[60] J. B. Hough, M. Krishnapur, Y. Peres, B. Virág, et al. Determinantal processes and independence. *Probability surveys*, 3:206–229, 2006.

[61] W.-L. Hsiao and K. Grauman. Creating capsule wardrobes from fashion images. In *CVPR*, 2018.

[62] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[63] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[64] Z. Ji, K. Xiong, Y. Pang, and X. Li. Video summarization with attention-based encoder-decoder networks. *arXiv preprint arXiv:1708.09545*, 2017.

[65] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

[66] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *European Conference on Computer Vision*, pages 293–306. Springer, 2008.

[67] T. Kanade and M. Hebert. First-person vision. *Proceedings of the IEEE*, 100(8):2442–2453, 2012.

[68] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *ICLR*, 2018.

[69] K. KAUST. End-to-end, single-stream temporal action detection in untrimmed videos.

[70] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1857–1865, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/kim17a.html`.

[71] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017. doi: 10.1145/3065386. URL `http://doi.acm.org/10.1145/3065386`.

[74] S. K. Kuanar, K. B. Ranga, and A. S. Chowdhury. Multi-view video summarization using bipartite matching constrained optimum-path forest clustering. *IEEE Transactions on Multimedia*, 17(8):1166–1173, 2015.

[75] A. Kulesza and B. Taskar. Structured determinantal point processes. In *NIPS*, 2010.

[76] A. Kulesza and B. Taskar. Learning determinantal point processes. 2011.

[77] A. Kulesza and B. Taskar. Learning determinantal point processes. 2011.

[78] A. Kulesza and B. Taskar. Learning determinantal point processes. *arXiv:1202.3738*, 2011.

[79] A. Kulesza, B. Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[80] A. Kulesza, B. Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[81] Y. LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[82] Y. J. Lee and K. Grauman. Predicting important objects for egocentric video summarization. *International Journal of Computer Vision*, 114(1):38–55, 2015.

[83] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1346–1353. IEEE, 2012.

[84] F. Li, Y. Fu, Y.-H. Dai, C. Sminchisescu, and J. Wang. Kernel learning by unconstrained optimization. In *Artificial Intelligence and Statistics*, pages 328–335, 2009.

[85] N. Li, D. Xu, Z. Ying, Z. Li, and G. Li. Searching action proposals via spatial actionness estimation and temporal path inference and tracking. In *Asian Conference on Computer Vision*, pages 384–399. Springer, 2016.

[86] P. Li, Y. Guo, and H. Sun. Multi-keyframe abstraction from videos. In *2011 18th IEEE International Conference on Image Processing*, pages 2473–2476. IEEE, 2011.

[87] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2855–2862. IEEE, 2012.

[88] X. Li, B. Zhao, and X. Lu. Mam-rnn: multi-level attention model based rnn for video captioning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2208–2214. AAAI Press, 2017.

[89] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[90] Y. Li, L. Wang, T. Yang, and B. Gong. How local is the local diversity? reinforcing sequential determinantal point processes with dynamic ground sets for supervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–167, 2018.

[91] Z. Lin, A. Khetan, G. C. Fanti, and S. Oh. Pacgan: The power of two samples in generative adversarial networks. *NIPS*, 2018.

[92] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3209–3216. IEEE, 2011.

[93] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016.

[94] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016.

[95] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (CelebA) dataset. *Retrieved August*, 15, 2018.

[96] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2721, 2013.

[97] T. Lucas, C. Tallec, J. Verbeek, and Y. Ollivier. Mixed batches and symmetric discriminators for gan training. *ICML*, 2018.

[98] Y. Luo, L.-F. Cheong, and A. Tran. Actionness-assisted recognition of actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3244–3252, 2015.

[99] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1903, 2016.

[100] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542. ACM, 2002.

[101] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542. ACM, 2002.

[102] O. Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

[103] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, pages 1–10, 2017.

[104] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised video summarization with ad-

versarial lstm networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[105] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised video summarization with adversarial lstm networks. In *CVPR*, 2017.

[106] V. A. Malyshev and A. M. Vershik. *Asymptotic combinatorics with application to mathematical physics*, volume 77. Springer Science & Business Media, 2012.

[107] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015. URL `http://dblp.uni-trier.de/db/journals/corr/corr1511.html#MathieuCL15`.

[108] K. Matsuo, K. Yamada, S. Ueno, and S. Naito. An attention-based activity recognition for egocentric video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[109] J. Meng, S. Wang, H. Wang, J. Yuan, and Y.-P. Tan. Video summarization via multi-view representative selection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1189–1198, 2017.

[110] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, 2017.

[111] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[112] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL `http://arxiv.org/abs/1411.1784`.

[113] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018.

[114] A. Montes, A. Salvador, S. Pascual, and X. Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv preprint arXiv:1608.08128*, 2016.

[115] T. Nguyen, T. Le, H. Vu, and D. Phung. Dual discriminator generative adversarial nets. In *NIPS*, 2017.

[116] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

[117] W. OBILE. Ericsson mobility report, 2016.

[118] W. OBILE. Ericsson mobility report, 2016.

[119] K. Ogaki, K. M. Kitani, Y. Sugano, and Y. Sato. Coupling eye-motion and ego-motion features for first-person activity recognition. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, June 2012. doi: 10.1109/CVPRW.2012.6239188.

[120] S.-H. Ou, C.-H. Lee, V. S. Somayazulu, Y.-K. Chen, and S.-Y. Chien. On-line multi-view video summarization for wireless video sensor network. *IEEE Journal of Selected Topics in Signal Processing*, 9(1):165–179, 2015.

[121] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1029–1038, 2016.

[122] V. Y. Pan and Z. Q. Chen. The complexity of the matrix eigenproblem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 507–516. ACM, 1999.

[123] R. Panda and A. R. Chowdhury. Multi-view surveillance video summarization via joint embedding and sparse optimization. *IEEE Transactions on Multimedia*, 2017.

[124] R. Panda, A. Dasy, and A. K. Roy-Chowdhury. Video summarization in a multi-view camera network. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2971–2976. IEEE, 2016.

[125] R. Panda, N. C. Mithun, and A. Roy-Chowdhury. Diversity-aware multi-video summarization. *IEEE Transactions on Image Processing*, 2017.

[126] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[127] Y. Peng and C.-W. Ngo. Clip-based similarity measure for query-dependent clip retrieval and video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(5):612–627, 2006.

[128] G. Peyré, M. Cuturi, et al. Computational optimal transport. Technical report, 2017.

[129] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora. Compact cnn for indexing egocentric videos. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[130] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[131] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[132] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.

[133] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1060–1069, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `http://proceedings.mlr.press/v48/reed16.html`.

[134] K. Regmi and A. Borji. Cross-view image synthesis using conditional gans.

[135] K. Regmi and A. Borji. Cross-view image synthesis using geometry-guided conditional gans. *CoRR*, abs/1808.05469, 2018.

[136] E. Richardson and Y. Weiss. On gans and gmms. *arXiv preprint arXiv:1805.12462*, 2018.

[137] T. Robinson and F. Fallside. A recurrent error propagation network speech recognition system. *Computer Speech & Language*, 5(3):259–274, 1991.

[138] M. Rochan and Y. Wang. Video summarization by learning from unpaired data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7902–7911, 2019.

[139] M. Rochan, L. Ye, and Y. Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 347–363, 2018.

[140] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 1, page 3, 2009.

[141] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.

[142] A. Sharghi, J. S. Laurel, and B. Gong. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. *arXiv preprint arXiv:1707.04960*, 2017.

[143] A. Sharghi, A. Borji, C. Li, T. Yang, and B. Gong. Improving sequential determinantal point processes for supervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 517–533, 2018.

[144] K. Shmelkov, T. Lucas, K. Alahari, C. Schmid, and J. Verbeek. Coverage and quality driven training of generative image models. 2018.

[145] S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[146] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.

[147] P. Smolensky. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, pages 194–281. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X. URL `http://dl.acm.org/citation.cfm?id=104279.104290`.

[148] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, 2015.

[149] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[150] B. Soran, A. Farhadi, and L. G. Shapiro. Action recognition in the presence of one egocentric and multiple static cameras. In D. Cremers, I. D. Reid, H. Saito, and M. Yang, editors, *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part V*, volume 9007 of *Lecture Notes in Computer Science*, pages 178–193. Springer, 2014. doi: 10.1007/978-3-319-16814-2_12. URL https://doi.org/10.1007/978-3-319-16814-2_12.

[151] A. Srivastava, L. Valkoz, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in GANs using implicit variational learning. In *NIPS*, 2017.

[152] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[153] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning*, pages 843–852, 2015.

[154] J. Stanley. Police body-mounted cameras: With right policies in place, a win for all. *New York: ACLU*, 2013.

[155] Y.-C. Su and K. Grauman. Detecting engagement in egocentric video. In *European Conference on Computer Vision*, pages 454–471. Springer, 2016.

[156] A. Swartz. Gopro posts record fourth-quarter sales but stock falls 15 percent on poor outlook, 2015. URL https://www.bizjournals.com/sanjose/news/2015/02/05/gopro-shares-soar-after-selling-2-4-million.html.

[157] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

[158] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

[159] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[160] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[161] D. Tran, R. Ranganath, and D. M. Blei. Hierarchical implicit models and likelihood-free variational inference. *NIPS*, 2017.

[162] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[163] A. Vershik. *Asymptotic Combinatorics with Applications to Mathematical Physics: A European Mathematical Summer School held at the Euler Institute, St. Petersburg, Russia, July 9-20, 2001*. Springer, 2003.

[164] T. H. Vu, A. Dang, L. Dung, and J.-C. Wang. Self-gated recurrent neural networks for human activity recognition on wearable devices. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 179–185. ACM, 2017.

[165] L. Wang, Y. Qiao, X. Tang, and L. Van Gool. Actionness estimation using hybrid fully

convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2016.

[166] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua. Event driven web video summarization by tag localization and key-shot identification. *IEEE Transactions on Multimedia*, 14(4):975–985, 2012.

[167] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[168] C. Xiao, P. Zhong, and C. Zheng. Bourgan: Generative networks with metric embeddings. *NIPS*, 2018.

[169] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *AAAI*, pages 1618–1625, 2017.

[170] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4633–4641, 2015.

[171] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4633–4641, 2015.

[172] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Video description generation incorporating spatio-temporal features and a soft-attention mechanism. *arXiv preprint arXiv:1502.08029*, 2015.

[173] T. Yao, Y. Li, Z. Qiu, F. Long, Y. Pan, D. Li, and T. Mei. Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and dense-captioning events in videos.

[174] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017.

[175] R. Yonetani, K. M. Kitani, and Y. Sato. Ego-surfing first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5445–5454, 2015.

[176] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[177] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arxiv 1805.08318*, 2018.

[178] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1059–1067, 2016.

[179] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.

[180] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1059–1067. IEEE, 2016.

[181] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[182] B. Zhao, X. Li, and X. Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 863–871. ACM, 2017.

[183] B. Zhao, X. Li, and X. Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 863–871. ACM, 2017.

[184] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. *ICCV, Oct*, 2, 2017.

[185] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[186] K. Zhou, Y. Qiao, and T. Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[187] K. Zhou, T. Xiang, and A. Cavallaro. Video summarisation by classification with deep reinforcement learning. *arXiv preprint arXiv:1807.03089*, 2018.

[188] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[189] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, X. Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, volume 2, page 8, 2016.